Computer Networks 52 (2008) 2677–2689

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

A stratified traffic sampling methodology for seeing the big picture

Stênio Fernandes [a,c], Carlos Kamienski [a,b], Judith Kelner [a], Dênio Mariz [d], Djamel Sadok [a,*]

[a] Universidade Federal de Pernambuco, Centro de Informatica, Cidade Universitaria, 50732-970 Recife, PE, Brazil
[b] Universidade Federal do ABC, Santo André, SP, Brazil
[c] Centro Federal de Educação Tecnológica de Alagoas, Maceió, AL, Brazil
[d] Centro Federal de Educação Tecnológica da Paraíba, João Pessoa, PB, Brazil

ARTICLE INFO

Article history:
Received 25 September 2006
Received in revised form 8 April 2008
Accepted 16 May 2008
Available online 5 June 2008

Responsible Editor: S. Kasera

Keywords:
Traffic sampling
Measurement
Traffic engineering

ABSTRACT

This work explores the use of statistical techniques, namely stratified sampling and cluster analysis, as powerful tools for deriving traffic properties at the flow level. Our results show that the adequate selection of samples leads to significant improvements allowing further important statistical analysis. Although stratified sampling is a well-known technique, the way we classify the data prior to sampling is innovative and deserves special attention. We evaluate two partitioning clustering methods, namely clustering large applications (CLARA) and *K*-means, and validate their outcomes by using them as thresholds for stratified sampling. We show that using flow sizes to divide the population we can obtain accurate estimates for both size and flow durations. The presented sampling and clustering classification techniques achieve data reduction levels higher than that of existing methods, on the order of 0.1% while maintaining good accuracy for the estimates of the sum, mean and variance for both flow duration and sizes.

1. Introduction

Traffic monitoring is seen as an important tool for engineering networks of varying sizes [21]. Characterizing and understanding the traffic mix enables Internet Service Providers (ISPs) to devise new business and settlement models. Recently, the pie has been shifting from transport to content services and understandably many sought a piece of the new opportunities. Efficient measurement strategies became an essential tool contributing to a number of tasks some of which were unforeseen until today. Traffic measurement has been used for identifying anomalous behavior (such as unexpected high traffic volumes due to denial of service attacks, routing problems, etc.), for designing and validating new traffic models, for adequate planning of seasonal activities, such as upgrading network capacity, which allows introducing new services or the adoption of usage-based and content-based pricing.

Commonly used flow monitoring tools (such as Cisco's NetFlow [2] and Juniper's JFlow [17]) and packet level capture tools suffer from lack of scalability relative to link capacity. Monitoring high-speed links, such as OC-48 and OC-192 yields huge data volumes. Currently both CPU and storage capacities remain limiting factors. As link capacity and the number of flows grow, maintaining individual counters for each individual flow traversing a router becomes computationally and/or economically cumbersome to keep up with [1,5,6]. In order to address this issue, sampling has been used a proven statistical strategy for dealing with high volumes of network traffic data [3]. It has the advantage of lowering the processing cost, the storage and hardware requirements in a network monitoring infrastructure.

Like a quality control engineer, a network engineer needs to collect data at diverse time frames and places. Therefore, to make this process faster as well as accurate, one should rely on analyzing only a small, although relevant, subset of the network traffic data. This sample is then

* Corresponding author. Tel./fax: +55 81 2126 8954.
E-mail addresses: stenio@gprt.ufpe.br (S. Fernandes), cak@gprt.ufpe.br (C. Kamienski), jk@gprt.ufpe.br (J. Kelner), denio@gprt.ufpe.br (D. Mariz), jamel@gprt.ufpe.br (D. Sadok).

1389-1286/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2008.05.011

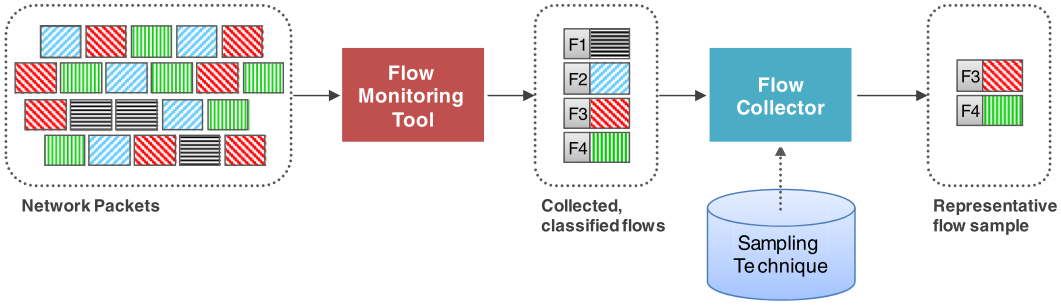

Fig. 1. Traffic monitoring, flow classification and filtering.

analyzed and a statistical quality verdict is given. In other words, one only needs to observe a small part of the traffic to visualize the big picture.

Fig. 1 shows how sampling can be used as a flow filter capable of selecting a representative set of flows only. By representative we mean those when selected, will give us an accurate view of the traffic mix. We show in this work that the classification strategy – also known as clustering – is primordial to achieve reduction of stored data but yet keeping its representativeness.

Sampling may be applied during the capture (packet level) or after data summarization (flow level). Fig. 2 depicts the envisioned benefits of using sampling techniques for flow traffic data management. A typical scenario is made of a router that captures packets and builds flows, a collector that retrieves and stores flow records and an application with a Graphical User Interface (GUI) that generates reports out of the stored data. The numbers indicate the possible sources of improvements when deploying sampling techniques at different components of the network management apparatus. On-line sampling may be performed directly by routers at the time they are capturing packets and building flows, whereas off-line sampling is performed after flows are built by the router and transmitted to the flow collector. If off-line sampling is used, optimizations may be obtained in the reduced storage requirements for the collector (#3), reduced data communication from the collector to the reporting application (#4) and reduced memory and processing requirements at the reporting application (#5). On the other hand, if on-line sampling is used, optimizations may be obtained at the reduced processing and memory requirements for the router (#1) and reduced data communication from the router to the collector (#2).

The processing and memory requirements will certainly be reduced at the router, since they preserve all flows in memory until they are flushed to the collector and the memory is cleared. If sampling is deployed, the router may detect that a given flow does not need to be stored, so that the allocated memory space for that particular flow can be released and the processing work of the remaining flows will be performed faster.

A variety of sampling strategies have been proposed recently for optimizing the packet selection process (for flow accounting) [18–20] and flow selection (for statistical analysis of the original traffic). Sampling techniques may be divided into systematic, random and stratified sampling [4,24]. The simplest sampling process is uniform $1/N$ sampling (a type of systematic sampling), where one of every N packets is retained. Although this technique has been largely used, *e.g.* by CISCO's Netflow in low-speed links, it does not always present adequate results since it has been shown that IP flows usually have heavy-tailed distributions for packets and bytes [6]. In such a situation, the selected packets from the sampling process will be biased by the huge number of small packets and just a few will be selected from the tail, *i.e.*, the bigger packets. Sampling may also be target dependent where for example only flows above a given size or duration are retained for charging purposes or selective of given services such as VoIP and

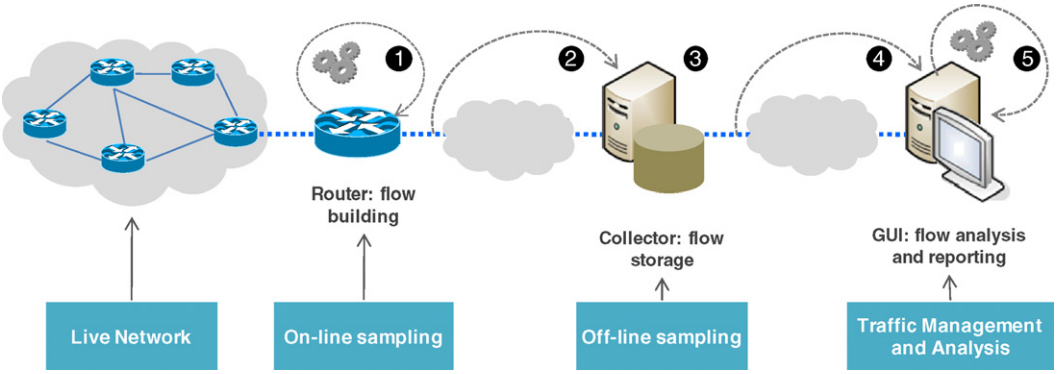

Fig. 2. Envisioned benefits of sampling.

other P2P file sharing activities for content-based charging [7]. Some sampling strategies, however, rely on flow information as they account only for relatively large flows, also leading to a biased sampling process due to the unbalanced number of selected small flows (also known as "mices") and bigger flows (also known as "elephants") [25].

The main contribution of this work is the introduction of a new flow classification methodology which applies two well studied statistical techniques, namely stratified sampling and cluster analysis, in order to reduce the amount of collected traffic that is necessary for characterizing the traffic flows. The role of stratified sampling is the intelligent grouping of the elements of the population in a number of strata (a set of samples), according to a key information from each flow (*e.g.* flow size) and in such a way that each stratum contains only flows with certain characteristics or limits (*e.g.* flow size greater than 500 bytes and lesser than 1000 bytes). For the traffic analysis, the samples are then taken from each subset or stratum, instead from the original population. We show that if we do our flow classification right, then important gains are achieved, as discussed later in this paper.

There may be different ways of classifying flows into groups and in this paper we present the results for two different approaches. Our first approach consists on defining the limits of the strata manually, according to an empirical observation. However, since the limits between strata have a significant importance for stratified sampling and the traffic profile keeps changing in the future [22], it becomes important to automate this step rather than relying on human knowledge. Then, we turn to investigating the effects of finding these limits automatically by using cluster analysis techniques. We show the effectiveness of using stratified sampling and cluster analysis as a tool for describing traffic behavior at a flow level. The measurement process is reduced to the capture and storage of only a small fraction of the original number of flows. Compared to existing methods, our proposed stratified sampling and flow classification methodology achieves higher reduction levels on the sample size with good accuracy in the estimates for the sum, mean and variance for flow duration and flow size.

The remainder of this paper is structured as follows: Section 2 reviews related work where sampling is used for traffic monitoring. In Section 3, we present the necessary background on stratified sampling and cluster analysis, which will be important to better understand the rest of the paper. Section 4 presents the data classification methodology we are proposing and also discusses the mechanism for evaluating its performance using real traces. Section 5 discusses the main results obtained and compares our strategy with another one, namely smart sampling. In Section 6, we discuss some choices that are involved in an actual deployment of stratified sampling for gathering traffic flow information. Finally, Section 7 concludes this work.

2. State of the art

Packet and flow sampling have drawn a great deal of research in the attempt to estimate traffic profiles. Some works particularly focused on router packet sampling whereas others chose to define resources necessary for building the measurement infrastructures [10,11,13,18].

A close approach to ours is that of Duffield et al. [5,8]. The authors pointed to the use of size dependent sampling instead of uniform sampling. In what they called *Smart* or *Threshold Sampling*, an object of size x is selected according to a sampling probability function $p_z(x) = \min\{1, x/z\}$. Therefore, flows of size less than a threshold z are selected with probability x/z whereas flows with size equal to or greater than z are always selected. The authors showed that they could control the volume of samples and the variance of an estimator (*e.g.*, sum of sampled sizes) by using threshold sampling. In a further study [9], they also presented an approach capable of inferring the probability distribution for the number and length of flows in the original Internet traffic. Their results are based on flow information built from a sampled set of packets.

Honh and Veicht [15] looked at some theoretical results for the problem of recovering traffic statistics, higher than first moment order, using sampling. They named their approach, *inverted sampling* and applied it to both packet and flow filtering. The authors also defined three statistical information recovery levels within their model: the packet level dealt with the spectral density for packet arrival; the flow level examined per flow packet distribution; and finally an internal level to the flows that determined the packet arrival average rate for a given flow. They discovered that their sampling technique at the flow level provided excellent results even with thinning probabilities on the order of 1%. Amazingly, the Inverted Sampling technique could recover detailed characteristics of the raw data traffic such as its spectrum or the distribution of flow size. Under our proposed stratified sampling and flow classification methodology, we achieve even higher reduction levels, on the order of 0.1% with good accuracy in the estimates of the sum, mean and variance for flow duration and sizes.

Estan and Varghese [11] proposed two scalable algorithms for identifying large (also known as elephant) flows, namely, *Sample and Hold* and *Multistage Filters*. Their sampling strategies only maintained records for large flows they called Elephants, hence reducing considerably the storage requirements. More importantly, Estan et al. [12] pointed out a number of limitations in the NetFlow architecture [1]. They particularly drew the attention to the maximum number of records that Netflow can classify and that a saturation level may quickly be reached when faced with a large burst of flows. A second NetFlow architectural problem is seen to be traffic congestion near the collector, a special server used as part of this architecture to collect all the sampled data. In addition, information loss on the path from a router to the collector can be extremely harmful to the analysis itself [10]. Finally, NetFlow's sampling strategy is static and does not adapt to the sampled traffic [12]. Intuitively one would argue for the increase of the sampling rate when aggregate traffic is low to obtain a higher resolution of the traffic profile. On the other hand, this rate must be lowered when the monitored traffic increases to avoid overflow in the measurement architecture. To this respect, these authors also proposed a new dynamic rate sampling for NetFlow they named *Adaptive NetFlow* (ANF).

In a separate piece of work, Kompella and Estan [18] proposed a novel flow measurement architecture called *Flow Slices*. Their elegant solution combines features from previous research work such as threshold sampling [5], adaptive NetFlow [12] and sample and hold [11].

3. Background

3.1. Stratified sampling

Stratification has proven itself as a technique capable of leading to the discovery of a population's statistical features by only looking a subset of it. That is, choosing only a few elements as a representative subset of the entire sample set. Under stratified sampling [4] a heterogeneous population is divided into isolated subpopulations that present certain level of homogeneity. It is also common to use some *a priori* knowledge to divide a population with N elements into $N_1,\ldots,N_L$ groups or strata. First, one should find the amount of the objects within each stratum is $n_1,\ldots,n_L$, respectively. Next, a sample is selected from each stratum or subgroup.

Stratified sampling can be classified into uniform, proportional (or Bowley), and optimal. The key step here is to determine the size of each stratum. Unlike the uniform stratification where all strata have the same size, proportional stratification maintains some proportionality among the number of elements in each stratum, whereas the optimal approach considers in addition to the stratum size the variability among its elements. We chose to use optimal stratified sampling without reposition of selected elements, meaning that each flow must be selected no more than once. Hence the strata are created according to both their size and variability.

We assume a population with N elements to be divided into L strata. The aim here is to look for n elements that need to be sampled in order to determine a statistical characteristic, such as, the mean flow size or its mean duration. The sample size, n, should be calculated from:

$$n \geqslant \frac{k^2AN - k^2B(N-1)}{\varepsilon^2(N-1) + k^2B}, \tag{1}$$

where k is the quartile $(1-\alpha)$ of the standard normal distribution and ε is the precision error. The terms A and B are given by Eqs. (2) and (3), where N_h and σ_h are the number of elements and the standard deviation within a given stratum h, respectively

$$A = \frac{\sum N_h\sigma_h^2}{\sum N_h} - \left(\frac{\sum N_h\sigma_h}{\sum N_h}\right)^2, \tag{2}$$

$$B = \frac{\sum N_h\sigma_h^2}{\sum N_h}. \tag{3}$$

Neyman [1] established a criterion for the distribution of the sample elements as long as they maintain a minimum variance within a stratum. It states that the number of elements within a given stratum, n_h, considering a sample of size n, should be followed by:

$$n_h = n\frac{N_h\sigma_h}{\sum N_h\sigma_h}. \tag{4}$$

3.2. Cluster analysis

Cluster analysis (CA) refers to a number of methods for grouping similar objects into categories. In general, clustering techniques have been mainly used for statistical data analysis in several research fields, such as data mining, machine learning, etc.

CA is an important step in our sampling methodology especially as we are seeking to extract traffic information from multivariables, that is, flows with size, duration and other attributes. The output of this technique is the basis that forms our *a priori* knowledge on traffic that stratified sampling is going to explore. Clustering can therefore be used to identify similarities between the analyzed flows and classify these into groups that are highly dissimilar from each other [14]. Such grouping not only depends on the data at hand but also on how good we formally define the similarity concept. This can mathematically be modeled as a distance between two objects. Please note that in multivariable clustering, one should always rely on normalization, which is a process to remove the effect of a variable into another to decrease its major influence on the distance metric. Also note that CA is also capable of identifying odd values as they tend to be exceptionally distant from most groups. A key element to the CA methodology is the adequate choice of the proximity concept definition. The literature is full of mathematical formulas expressing distances between multivariable objects. A general formulation is given by Minkowski [16]:

$$d_p(x_i, x_j) = \left(\sum_{k=1}^{d} |x_{i,k} - x_{j,k}|^p\right)^{1/p}, \tag{5}$$

where d is the multivariable object dimension. The well-known Euclidean and Manhattan distances are special cases of the Minkowski distance with $p = 2$ and $p = 1$, respectively.

Once we have our distance defined, we need to establish the CA technique we are going to use. There is a wide range of CA algorithms available in the literature although no canonical classification. They do however differ according to the data being analyzed, the clustering criteria they use and also the mechanism they use to approach the solution including statistical, fuzzy logic and other strategies. Unlike other existing studies where clustering can be based on density, probabilistic, monothetic (decisive) or polithetic (open), we opted for a flexible clustering technique that can adopt a hierarchical or partitioned vision of the flows. In this work we examine two clustering techniques, namely, *K*-means and CLARA [16].

Both popular and simple, *K*-means offers a reasonably good clustering technique for our problem. We first look for k *centroids* or *geometrical centers*, one for each group. Next, *K*-means assigns each of the flows to the group of the nearest *centroid* using our previously defined Minkowski multivariable distance concept, with $p = 2$ (*i.e.*, the Euclidean distance). Once done with all the flows, we need to review our initial decision by selecting new k *centroids*. This process is continued until the proposed k *centroids* stop changing and we have our k stable clusters. We need to say something regarding how *K*-means compares the

effectiveness of a newly established cluster around a *centroid* with that previously obtained when using a different *centroid*. *K*-means uses a fitness test given by

$$J = \sum_{j=1}^{k} \sum_{i=1_i}^{n} d_{j,p}, \quad (6)$$

where $d_{j,p} = \|x_i^{(j)} - c_j\|^p$ is the distance metric between $x_{i,j}$, a given element of group j, and c_j, the centroid of group j. It gives us a measure of the distance of the n elements from their respective cluster centers.

The second clustering method we look into is known as CLARA (Clustering LARge Applications), which is an improved implementation of PAM (Partitioning Around Medoids). PAM is considered a more robust version of *K*-means, as it minimizes the sum of dissimilarities instead of the sum of squared Euclidean distances. Also, PAM is used typically when we want to create clusters around typical objects (flows) known as *medoids*. Once we establish the *medoids* using iterative selection, all flows are then assigned to the group with the nearest *medoid*. PAM is $O(n^2)$ with asymptotic complexity to memory use turning it unsuitable as is for the clustering of large data sets.

CLARA is a specially devised PAM to deal with this problem, as it exhibits a linear execution time and memory use characteristics, having time complexity $O(n)$. CLARA uses mainly two single stages. First a sample of predefined length is retrieved from the data set and distributed over the k groups. Successive *medoids* are examined in an attempt to find the one with the shortest distance to flows from the sample in what is known as the build phase. Next, the most representative elements are swapped in order to lower the average distance between objects in a group. This second phase is known as the Swap phase. More details on both *K*-means and CLARA may be obtained from [16].

4. Methodology

This work analyzes four different flow traces shown in Table 1. These were obtained from the national Brazilian academic network (RNP) at PoP-PE over a 34 Mbps link between the 15th and 19th of September 2004. We used a packet capturing tool that fed a flow identification module, where a flow is seen as a set of packets with the same 5-tuple: source IP address, destination IP address, port numbers for origin and destination and protocol [23].

Core to our approach is the way flow data has been stratified. We show the importance of our methodology as it offers a 10-fold gain over existing works in some of the analyzed scenarios. Not only the use of stratifying is beneficial but similarly the way this is applied is of utmost relevance. Clearly there are a number of ways we may think of when grouping the data. These include classifying flows according to their sizes, durations, transmission rates, etc. In choosing one of these schemes, it is then important to be able to obtain as much flow statistical information as possible. For example, classifying flows according to sizes should also provide us with information on flow durations. The statistics should not be seen as exclusive with flow classification criteria as this process is merely seen as a step leading to the manipulation of a smaller but yet representative data sets or population. We also show that the number of such data groups or strata is an important study factor.

Table 1
Characteristics of the traces used in the analysis

Name	Date	Time (h)	Volume (GB)	# Flows
Trace #1	September 15, 2004	08–12	28	7,013,744
Trace #2	September 17, 2004	08–12	38	7,497,991
Trace #3	September 18, 2004	14–18	24	4,086,476
Trace #4	September 19, 2004	14–18	63	6,571,586

We propose a data classification methodology with the following seven steps:

1. *Deciding which variables we need to observe* – there is a wide range of flow attributes that are often interesting to monitor. Examples of these include number of packets, packet minimal/average/maximal sizes, transport and application protocol used, etc. Without loss of generality, we limited the variables monitored in this work to: (a) flow duration in seconds; and (b) flow size, defined as the amount of bytes in all the packets of a flow. Note that although we observed two distinct variables, only flow duration is presented as the stratifying variable, *i.e.* the variable used for classifying flows. Flow size was also tested as the stratifying variable when we considered manually established strata, as presented in Section 5.1, and the results were similar to flow duration. Therefore, since both variables perform adequately with clustering, for simplicity only flow duration is presented as the stratifying variable throughout the paper.

2. *Grouping flows and setting up strata limits* – this step refers to the process of finding out groups of flows related to each other and defining the upper limit of the stratifying variable (flow duration) for each strata. There may be different ways of classifying flows into groups and in this paper we present the results for two different approaches. Our first approach consists on defining the limits of the strata manually, with an exponential increase (*e.g.* 0.01 s, 0.1 s, 1 s, 10 s, 100 s, 1000 s, 10,000 s). This choice relies on an expert knowledge, *i.e.* an empirical observation that flows can be grouped into strata according to these limits. Our second approach investigates the effects of finding these limits automatically by using cluster analysis on a trace. Our goal is twofold. First, since the limits between strata have a significant importance for stratified sampling and the traffic profile keeps changing in the future, it becomes important to automate this step rather than relying on human knowledge. Second, intuitively we consider the possibility of improving the precision of the stratified sampling by choosing optimal limits between strata, aimed at finding sub-samples with less internal variability. In this work, we use both CLARA and *K*-means algorithms to identify meaningful groups according to flow duration thresholds.

3. *Choosing the number of strata* – the two approaches for grouping flows into strata (manual-based and clustering-based) require the prior knowledge of the number of strata to work with. Preliminary experiments showed that the use of more than 8 strata gives little gain in results

accuracy while leading to higher processing overhead. Consequently, we limit the number of strata to 2, 4, 6 and 8. Indeed, our main interest is on finding the minimum number of strata that gives us acceptable results.

4. *Establishing sample size* – we obviously seek to use the lowest sample size capable of representing the original population of flows. We tested four sample sizes, namely, 0.01%, 0.1%, 1% and 10% of the original population size. Sample sizes smaller than 0.01% were considered inadequate, since results reveal that with this size the sample is not representative in most cases. Also, sample sizes larger than 10% were also not used, since the reduction is not significant.

5. *Determining the number of elements per stratum*: the number of elements varies for different scenarios and in our case needed to be computed for all traces (trace #1 to #4, as shown in Table 1), approaches to grouping elements (manual and automatic with CLARA and *K*-means), number of strata (2, 4, 6, 8) and sample sizes (0.01%, 0.1%, 1% and 10%). We use the Neyman method where the sample sizes are chosen proportional to the products of the standard deviations and the stratum sizes [1].

6. *Establishing the achieved accuracy* – we adopted Monte Carlo Simulation as a validation tool to establish the accuracy achieved by each of scenarios considered in our work. This step works as follows: (a) consider the population of flows from a given trace; (b) separate the flows into each strata according to their limits; (c) for each stratum of the population, take randomly some elements to make up the stratum of the sample, based on step 5; (d) having the sub-sample for each stratum, compute sample flow size and duration using the estimator of the stratified sampling technique; (e) repeat this process 100 times and obtain the average flow size and duration for the sample. We used a 99% asymptotic confidence interval.

7. *Comparing population and sample metrics* – in order to find out whether a sample is representative of the population, the metrics of both must be compared. In other words, the average, sum and variance for flow duration and sizes of the samples must be a representative approximation of those of the population; otherwise the sample may not be used in place of the population. Although we observe the average, standard deviation and sum of flows durations and sizes, only average values are presented.

In addition to the proposed stratified sampling methodology, both uniform and smart sampling [5] have also been compared.

5. Results

5.1. Results for stratified sampling

Using the proposed stratified sampling technique we compare mean values for flow size and duration obtained for both the population and its samples. As an example, in Fig. 3 we compare the average of the samples for the mean of flow sizes with that of the original population. We have done this for each one of the four traffic traces.

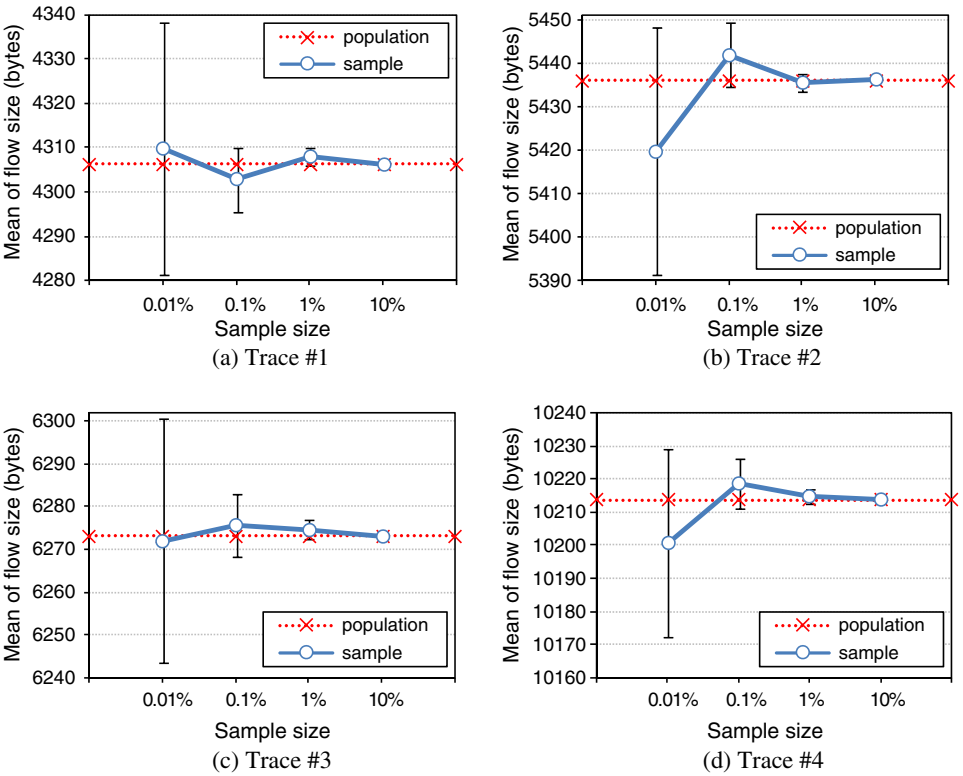

Fig. 3. Mean of flow size for different traces (using 8 strata).

Such comparison is important as it gives us a statistical guarantee for using a sample instead of the original population. As explained in Section 4, we first used some rule of thumb regarding knowledge of typical behavior of flow sizes and durations to set the limits between the strata. These results are presented in this section. In a second stage such limits were obtained automatically from the traces in an unsupervised manner, as we show in Section 5.2.

Under the intuitive approach, the results for the mean of flow size for our four traces, four sample sizes (0.01%, 0.1%, 1% and 10%) and with 8 strata are depicted in Fig. 3. For the samples, we also plotted the vertical bars of the 99% asymptotic confidence intervals obtained from 100 replications. A graphical analysis shows that the larger the sample size is, the more precise is the confidence interval, *i.e.*, its amplitude is smaller. The dotted line represents the actual real mean of the population computed separately for all the captured traces' flows. It can be seen that the population mean falls within all the established confidence intervals. In other words, the intervals built from the samples are representative and the samples may be used instead of the original captured flow population for representing the mean of flow sizes. For instance, in Fig. 3a the mean flow size for the population is 4306.3 bytes and for a 1% sample size is 4308 with a ±2.1 byte confidence interval.

This first result emphasizes the existing tradeoff between sample size and precision (including variability). On the one hand whenever a higher precision is needed, a larger sample size should be adopted, in order to ensure that any sample will be able to represent the population. On the other hand, if a lower precision is tolerated by a given analysis, then even sample sizes of 0.01% may be used while leading to lower requirements for the infrastructure in terms of processing, transmission and storage costs.

From Fig. 3, we also see that the results obtained for all four traces present similar behavior. Such similarity shows up not only for the mean of flow size obtained from of a given number of strata, but also for the flow size sum and variance for all the strata. Sum and variance are not shown in our results, since the latter is similar to the mean (or more precisely, the estimator of the mean is directly derived from the estimator of the sum) and the former brings no practical insights for ISPs, although bearing a great statistical importance. Therefore, due to space constrains throughout this paper only Trace #1 will be used (our choice was random). Its population size is 7,013,744 flows, and consequently the number of flows is: 701, 7013, 70,137 and 701,374 flows for sample sizes of 0.01%, 0.1%, 1% and 10%, respectively. Furthermore, the vertical bars representing the confidence intervals will be omitted in most results henceforth, since they have equivalent dimensions for a variety of different days, sample sizes and number of strata.

Fig. 4 shows results of both flow size and duration for Trace #1 while varying the number of strata used over different simulation experiments (in Fig. 3, only the results for 8 strata were shown). Three important observations can be made from Fig. 4. First, the use of a 0.01% sample size turns out to be insufficient, since its samples exhibit large variability as shown by both Fig. 4a and b. Second, more reliable results are obtained when the number of strata is at least six, as the confidence intervals become small and always include the population mean. On the other hand, the gain seen when using 8 strata instead of six is small, leading us to believe that no more than 8 strata are needed for most cases. Third, flow size and duration yielded similar results.

5.2. Finding automatic limits among the strata

In this section we take a closer look at the CLARA and *K*-means clustering techniques in the effort to finding more adequate limits between the strata. For each technique, we ran the clustering algorithms with four different numbers of groups: 2, 4, 6 and 8. In a typical case, clustering algorithms are capable of finding the "ideal" number of groups on their own. Once we had the elements (flows) allocated to their groups, both lower and upper limits of each group were found.

Table 2 presents the lower and upper limits yielded by CLARA and *K*-means when configured to work with two groups. Interestingly, we see that for both, the limits do not overlap for flow duration. For example, CLARA flows lasting less than 155 s were classified into group 1, whereas the others fell into group 2. Flow size limits overlap however between these two groups. For example, for

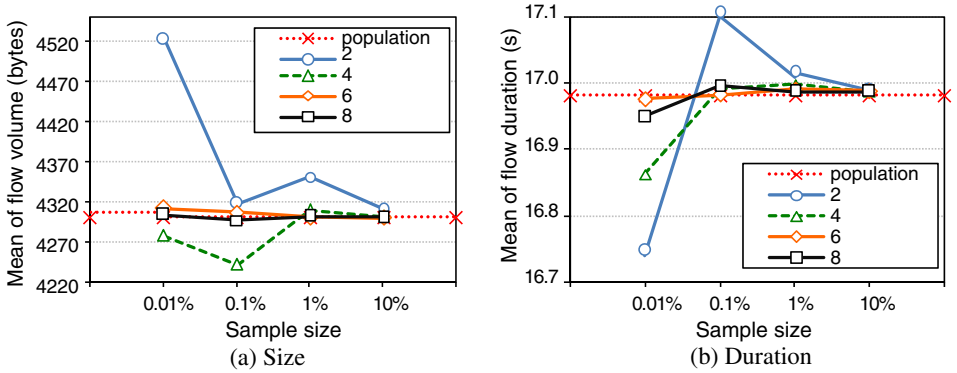

Fig. 4. Mean of flow size and duration (Trace #1).

K-means the first group includes flows from 23 bytes to 115 MB and the second one those ranging between 330 bytes and 426 MB. This leads for the fact that flows of 1 MB may fall in any or both of the two groups. We decided next to use only flow duration as the stratification variable as we were unable to establish clear strata limits for flow sizes.

Fig. 5 depicts the upper limits of the flow duration for 2, 4, 6 and 8 groups produced by CLARA and *K*-means, in a base 10 logarithmic scale. It can be observed that CLARA produces limits (except for the last group) approximately two magnitude levels lower than those obtained using *K*-means. This behavior is repeated throughout the four numbers of groups. In other words, *K*-means tends to put a high number of elements into the first stratum leaving fewer to the remaining strata. We will come back later to this point and show that it has a significant impact on the stratified results.

In a traffic analysis tool, we may of course think of using this to adapt dynamically the limits between strata based on the current characteristics of the flow records, but in a time scale significantly higher than dealing with individual flow records. In this paper, we limit ourselves to comparing clustering techniques with and without the automatic set-up of such limits. We will use the term "*expo*" when referring to the results of the stratified sampling produced by manually specified exponential limits.

5.3. Using stratification together with clustering

In Section 5.1 we manually configured the limits between contiguous strata using our prior knowledge of traffic and work experience. As explained in Section 4, now we consider a different approach for setting up the limits of strata, *i.e.* using the upper limits identified by the cluster analysis of Section 5.2.

Fig. 6 depicts the results for the mean of flow duration for 8 strata that are similar to those shown in Fig. 3. First we see that the stratified samples produced by CLARA and *K*-means are acceptable since all confidence intervals cover the population mean. More importantly and as expected, we see that CLARA outperforms *K*-means, *i.e.* that its limits produce more representative samples. This may be observed for precision (proximity of the population

Table 2
Lower and upper limits for 2 groups (*Trace #1*)

Group	Duration (s)				Size			
	CLARA		*K*-means		CLARA		*K*-means	
	Lower	Upper	Lower	Upper	Lower (B)	Upper (MB)	Lower (B)	Upper (MB)
1	0.0	155.0	0.0	3.3	23	24	23	115
2	155.0	83.8	3.3	83.8	56	426	330	426

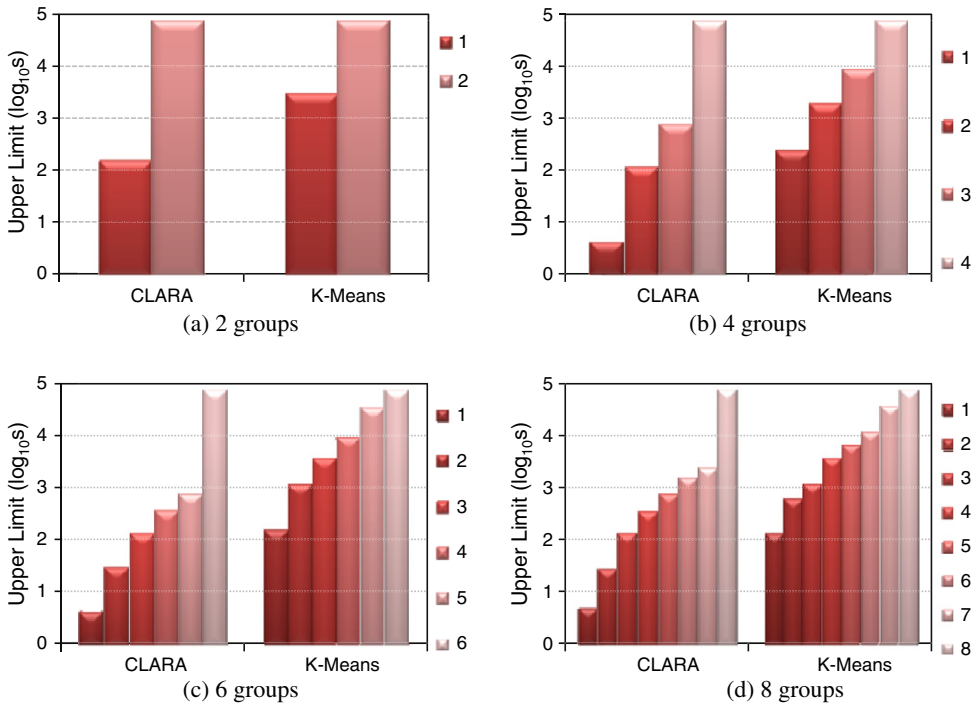

Fig. 5. Upper limits from the groups considering the flow duration (*Trace #1*).

mean) and for samples variability (size of the confidence interval).

We saw earlier that the upper limit of the first group of *K*-means is almost two levels of magnitude higher than CLARA leading *K*-means to classifying more elements into the first group. For example, for eight groups on the ***Trace #1***, CLARA yields an upper flow duration limit of 4.5 s and includes 5,005,776 flows, corresponding to 71.4% of the total number of flows in the trace. On the other hand, for *K*-means this limit is 156 s with as many as 6,863,446 flows, representing 97.9% of the flows therefore leading to a larger sample variance as longer flows are also included. Since higher strata (those with higher flow durations) have smaller sub-samples, the estimator for the mean of the stratified sampling will give a higher weight to these sub-samples. The smaller sub-samples of flows with longer durations together with their higher weight make the estimator too much dependent on the specific elements taken from those strata for the sub-sample. Hence, the estimator for *K*-means computes less precise and more variable means than the one for CLARA. Our first conclusion is that the samples produced by *K*-means are less representative of the population, as confirmed by Fig. 6.

Fig. 7 depicts the stratification results obtained with different number of strata (2, 4, 6 and 8) for CLARA, *K*-means and expo (manual assigned limits for strata). Please recall that the results presented in Fig. 6 refer to 8 strata, which represented the case with the most suitable number of strata we found. These results from Fig. 7 are also similar to those of Fig. 4, showing that adequate results are yielded when: (a) using 8 strata for 0.01% sample sizes; (b) at least 6 strata for 0.1% and 1% sample sizes; and (c) for 10% sample sizes even 2 strata yield reliable results (although the advantage of using sampling is diminished).

5.4. Smart sampling

We wanted to know next how our proposed sampling methodology did compare with smart sampling described in [5,10]. Fig. 8 shows how good each method estimates the mean of flow duration under different sample sizes. First, *smart sampling* was able to build even lower confidence intervals than CLARA (confidence intervals are not shown in Fig. 8 for the sake of improving the visual quality of the picture). This result was expected as it uses a sampling threshold that is automatically chosen for getting lower variability among elements within the sample. Second, as far as our experiments are concerned, *smart sampling* was not able to obtain estimates of the population mean as precise as stratified sampling. Mainly for sample sizes higher than 0.01%, CLARA obtained at least the same precision as *smart*. Comparing with *K*-means however, *smart* had a higher precision.

We also saw that processing smart was a very CPU intensive task, actually several times slower than stratified sampling. For example, using a 3.2 GHz Pentium 4 station, smart sampling took a couple of hours for processing a 1% sample size experiment for one trace with 100 replications, whereas stratified sampling took only a few minutes to perform the same task. This turns *smart sampling*'s usage

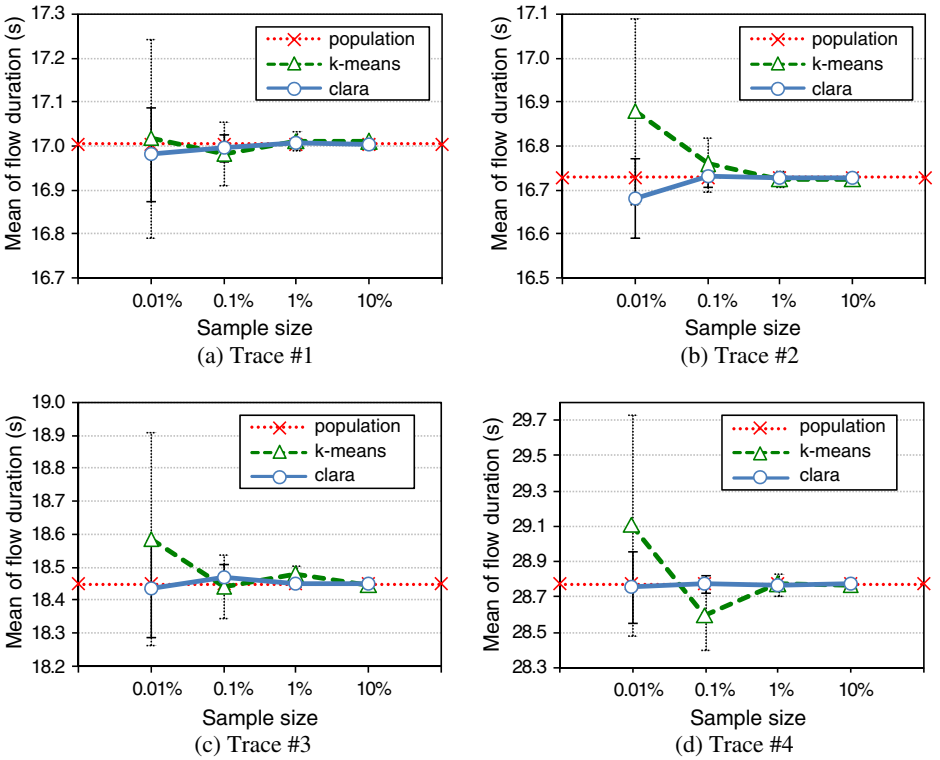

Fig. 6. Mean of flow duration for different traces (8 strata).

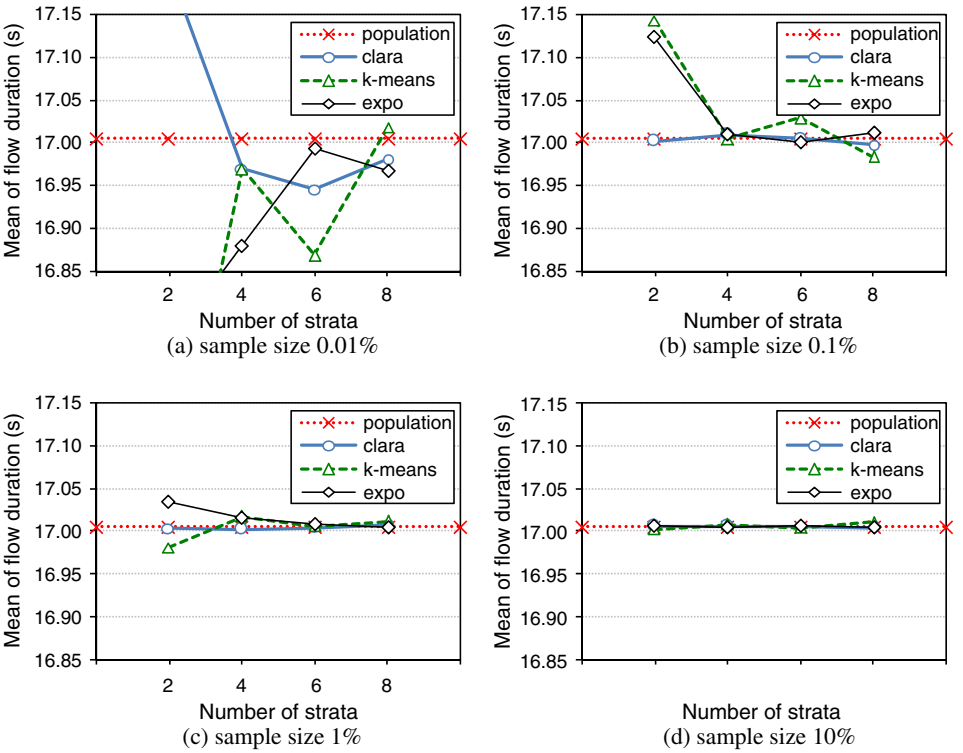

Fig. 7. User-defined limits (expo) vs. clustering (CLARA and *K*-means).

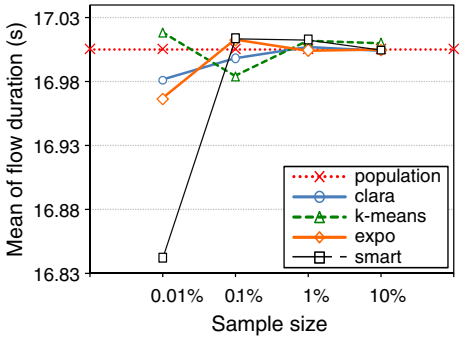

Fig. 8. Comparing our stratified sampling with the smart sampling proposal.

questionable in various scenarios, including real-time sampling within routers.

5.5. Computing the stratification gain

Uniform sampling features simplicity considering both processing and storage requirements, since only the samples need to be stored, whereas stratified sampling requires additional storage and processing for the strata information. Some benefits are needed to justify the use of the latter. Formally, the stratification gain for the mean g_{strat} is defined as the normalized difference between the bias of the mean of the estimate $\bar{y}$ obtained by uniform sampling *bu* and the one obtained using stratified sampling *bs*. The bias is the difference between the population mean and the mean computed for the sample. The stratification gain for the variance of the mean is defined in a similar way

$$g_{\text{strat}} = \begin{cases} \frac{(bu-bs)}{bu}, & \text{when } bu \geqslant bs, \\ -\frac{(bu-bs)}{bu}, & \text{when } bu < bs. \end{cases} \tag{7}$$

From Eq. (7) we see that the stratification gain is close to 0 when both biases (uniform and stratified sampling) have similar values. It approaches 1, when the stratified sampling yields a bias considerably smaller than that of the uniform sampling (*i.e.*, the gain is high) and approaches −1 otherwise.

Figs. 9 and 10 depict the stratification gain for both the mean and the variance for the CLARA and *K*-means methods, respectively. Fig. 11 shows the same for the manual (expo) stratification. Once more we see clear advantages for using CLARA with stratification gains close to 1 for 4, 6 and 8 strata, whereas as expected *K*-means suffers from higher variability. For example, with 2 strata the mean stratification gain is conservative for sample sizes smaller than 10% and averages 0.8 for the other number of strata. For the variance, the gain is around 0.85 for 4, 6 and 8 strata. CLARA outperforms the manual (expo) stratification, whereas *K*-means has even worse results. Note that our results still show that both clustering methods and the expo strategy for determining the strata limits yield considerable stratification gains.

These results also corroborate to the argument that there is no significant gain in increasing the number of

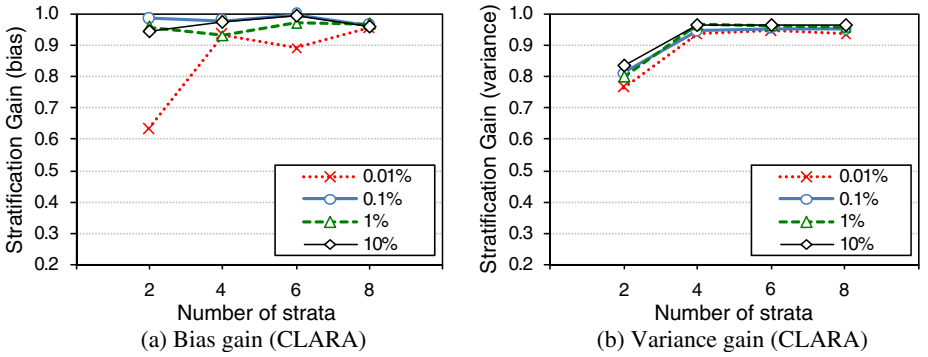

Fig. 9. Stratification gain for CLARA (Trace #1).

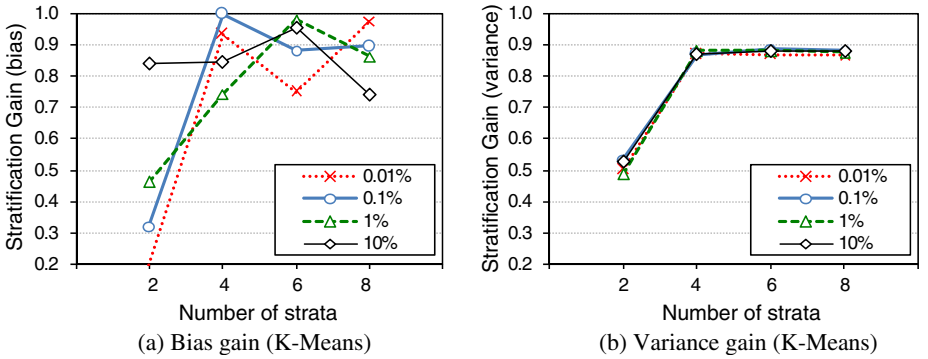

Fig. 10. Stratification gain for *K*-means (Trace #1).

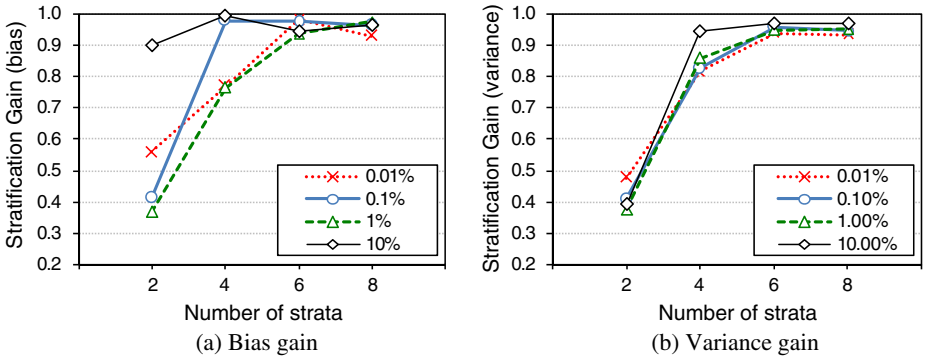

Fig. 11. Stratification gain for Expo (Trace #1): (a) bias gain and (b) variance gain.

strata. This may be observed with the shape (slope) of the curves. We also computed the stratification gain for the sum (of flow duration) and the results were similar to those in Figs. 9–11.

We also compared optimal and proportional stratification allocation techniques. Work in [24] concluded that for packet sampling using packet size as the stratification variable, no significant difference was observed. However, in our work, proportional stratification obtained extremely poor results. We attribute this to the heavy-tailed nature of traffic flows. We do not present the graphs comparing both approaches due to a lack of space.

6. Discussion

In this section we discuss some choices that are involved in any real deployment of stratified sampling for traffic flow information. These choices are stratification approach, sample size, number of strata, stratifying variable

and deployment scenario. The first choice is the stratification approach, and we analyzed three options: manual assignment with exponential strata limits, and dynamic assignment of strata limits with clustering methods CLARA and *K*-means. We found out in most evaluated scenarios that CLARA outperforms the other two approaches, so that it is considered the preferred choice for stratification approach in a real use of stratified sampling. The second and third choices refer to the sample size and number of strata respectively. Please notice that a real scenario of using stratified sampling for traffic measurement and/or analysis in an ISP probably would not involve changing the values of sample size and number of strata. We evaluated different possibilities for those parameters in order to understand the existing tradeoffs and to be able to choose the best option for every situation. Therefore, a specific value for those parameters should be chosen, preferably the one that provides the best precision for a given processing and storage cost. The larger the sample size and number of strata, the more precise the results. On the other hand, processing and storage costs also increase (each stratum is treated individually). Hence, the number of strata and the sample size should be kept as small as possible, for any given target precision. The results presented in this paper can help analysts configure the best values for some parameters. For the traces and scenarios analyzed in this paper, our results show that with CLARA stratification, 4 strata and 0.1% sample size represent a cost effective choice. The fourth choice refers to the stratifying variable, which in our paper was the flow duration. Results showed that stratification based on flow duration yields adequate results for both flow size and duration. Even though we were not able to test flow size as the stratification variable dynamic assigned strata limits, the results obtained with flow duration for CLARA outperformed those with flow size for manual assignment (as explained in Section 4 the latter were not presented in this paper). Therefore, our results corroborate the choice of flow duration as the stratifying variable.

Finally, the fifth choice is related to the deployment scenario. Our solution based on stratified sampling may be used in a non-real-time deployment for reducing the amount of flow traffic information that must be stored in a collector and transmitted to the definitive data storage. Since we need to know the whole trace in order to find the strata limits and to go through the stratification process, our solution is not prepared for performing sampling with incomplete traces. However, as large volumes of information are produced in a typical ISP each day, this should be regarded as an enabling tool for making it possible the storage, processing and analysis of traffic information.

The good piece of news is that with small changes in the method, our solution may be used in real-time by routers for reducing the amount of information to be kept in memory, as well as for transmitting to the collector. In order to do this, we need to perform a learning phase with a complete trace whereby we obtain the strata limits and the percentage of flows selected to belong to each sample stratum. For example, if the population stratum #1 has 1,000,000 flows and the sample stratum #1 has 1000, we learn that only 0.1% of flow of this stratum will be preserved for the sample. An important clarification here is that even though the sample size is pre-established as a percentage of the population size, in stratified sampling the same percentage will not be used in all strata. Therefore, we need the strata limits and the percentage of each sub-sample within the router to perform the real-time sampling. Routers may use this information like that. Whenever a new flow is identified (the data flow ended up), the percentage is used as a probability of keeping that flow in the sample. Periodically, the learning phase must be redone, *i.e.* new strata limits and percentages must be computed as traffic patterns change.

7. Conclusion

Traffic analysis remains a challenging task where any bit of help is welcome. In this work we looked at the application of stratified sampling and developed a methodology for it. We saw encouraging signs for its use when combined with CLARA-based clustering.

We show that stratified sampling and cluster analysis can be used as a tool for describing traffic behavior at a flow level. The measurement process achieves higher reduction levels on the sample size with good accuracy in the estimation of the sum, mean and variance for flow duration and flow size. We also discussed some choices that are involved in any real deployment of stratified sampling for traffic flow information. With small changes, our solution may be used in real-time by routers for reducing the amount of information to be kept in memory and traffic to the collector.

Understandably we remain further away from real-time schemes meanwhile for many applications such approaches are sufficient. Currently we are looking at the impact of sampling and clustering methods on the performance of networked applications identification techniques, such as those that perform blind inference, *i.e.* without looking at the packet payload, and also those that deal with anomaly traffic.

References

[1] P.D. Amer, L.N. Cassel, Management of sampled real-time network measurements, in: Proceedings of the 14th Conference on Local Computer Networks, 10–12 October 1989, pp. 62–68.
[2] CISCO NetFlow, <http://www.cisco.com/warp/public/732/Tech/nmp/netflow>.
[3] K.C. Claffy, G.C. Polyzos, H. Braun, Application of sampling methodologies to network traffic characterization, SIGCOMM Comput. Commun. Rev. 23 (4) (1993) 194–203.
[4] Cocharan, G. William, Sampling Techniques, third ed., John Wiley, New York, 1977.
[5] N. Duffield, C. Lund, M. Thorup, Learn more, sample less: control of volume and variance in network measurement, IEEE Trans. Inform. Theory 51 (5) (2005) 1756–1775.
[6] N. Duffield, C. Lund, M. Thorup, Flow sampling under hard resource constraints, in: Proceedings of the Joint international Conference on Measurement and Modeling of Computer Systems (New York, NY, USA, June 10–14, 2004), SIGMETRICS'04/Performance'04, ACM, New York, NY, 2004, pp. 85–96.
[7] N. Duffield, C. Lund, M. Thorup, Charging from sampled network usage. in: Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement (San Francisco, California, USA, November 01–02, 2001), IMW'01, ACM, New York, NY, 2001, pp. 245–256.
[8] N. Duffield, C. Lund, M. Thorup, Properties and prediction of flow statistics from sampled packet streams, in: Proceedings of the 2nd

ACM SIGCOMM Workshop on Internet Measurement (Marseille, France, November 06–08, 2002), IMW'02, ACM, New York, NY, 2002, pp. 159–171.
[9] N. Duffield, C. Lund, M. Thorup, Estimating flow distributions from sampled flow statistics, IEEE/ACM Trans. Netw. 13 (5) (2005) 933–946.
[10] N. Duffield, C. Lund, Predicting resource usage and estimation accuracy in an IP flow measurement collection infrastructure, in: Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement (Miami Beach, FL, USA, October 27–29, 2003), IMC'03, ACM, New York, NY, 2003, pp. 179–191.
[11] C. Estan, G. Varghese, New directions in traffic measurement and accounting, SIGCOMM Comput. Commun. Rev. 32 (4) (2002) 323–336.
[12] C. Estan, K. Keys, D. Moore, G. Varghese, Building a better NetFlow, SIGCOMM Comput. Commun. Rev. 34 (4) (2004) 245–256.
[13] S. Fernandes, T. Correia, C. Kamienski, D. Sadok, A. Karmouch, Estimating properties of flow statistics using bootstrap, IEEE MASCOTS 2004, October 2004.
[14] F. Hernández-Campos, A.B. Nobel, F.D. Smith, K. Jeffay, Understanding patterns of tcp connection usage with statistical clustering, in: Proceedings of the 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Atlanta, GA, September 2005, pp. 35–44.
[15] N. Hohn, D. Veitch, Inverting sampled traffic, IEEE/ACM Trans. Netw. 14 (1) (2006) 68–80.
[16] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Comput. Surveys 31 (3) (1999).
[17] JUNIPER JFlow IP Stats, <http://www.juniper.net/products/junose/105017.html>.
[18] R.R. Kompella, C. Estan, The power of slicing in internet flow measurement. in: Proceedings of the 5th ACM SIGCOMM Conference on internet Measurement (Berkeley, California, October 19–21, 2005), IMC'05, ACM, New York, NY, 2005, pp. 1–14.
[19] J. Laiho, K. Raivio, P. Lehtimaki, K. Hatonen, O. Simula, Advanced analysis methods for 3G cellular networks, IEEE Trans. Wireless Commun. 4 (3) (2005) 930–942.
[20] A. Lakhina, M. Crovella, C. Diot, Mining anomalies using traffic feature distributions, SIGCOMM Comput. Commun. Rev. 35 (4) (2005) 217–228.
[21] K. Papagiannakit, N. Taft, C. Diot, Impact of flow dynamics on traffic engineering design principles, in: INFOCOM 2004, Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 4, 7–11 March 2004, pp. 2295–2306.
[22] K. Papagiannaki, N. Taft, Z.-L. Zhang, C. Diot, Long-term forecasting of Internet backbone traffic: observations and initial models, in: INFOCOM 2003, Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 2, IEEE, 30 March–3 April 2003, pp. 1178–1188.
[23] R Development Core Team, R: A language and environment for statistical computing. R Foundation for Statistical Computing, 2007. <http://www.r-project.org>.
[24] T. Zseby, Stratification strategies for sampling-based non-intrusive measurements of one-way delay, Passive and Active Measurement Workshop Proceedings, April 2003.
[25] Liang Guo, I. Matta, The war between mice and elephants, in: Ninth International Conference on Network Protocols, 11–14 November 2001, 2001, pp. 180–188.

Stênio Fernandes received his Ph.D. in computer science from the Federal University of Pernambuco (Recife PE, Brazil) in 2006. He is currently a Professor of computer networks at the Federal Center for Education in Technology in Maceió, AL, Brazil. He also collaborates with GPRT, a research group in the areas of computer networks and telecommunications. His current research interests include traffic measurement, modeling and analysis, multimedia streaming on the Internet, and IPTV over P2P networks. He has been involved in several research projects funded by Brazilian agencies and also in partnership with telecom companies.

Carlos Kamienski received his Ph.D. in computer science from the Federal University of Pernambuco (Recife PE, Brazil) in 2003. He is currently an associate professor of computer networks at the Federal University of the ABC in Santo André SP, Brazil. His current research interests include policy-based management, traffic measurement and analysis and ambient networks. He has been involved in several research projects funded by Brazilian agencies and also in partnership with telecom companies.

Judith Kelner received her PhD from the Computing Laboratory at the University of Kent at Canterbury, UK in 1993. She is currently a Professor of Multimedia Systems at the Computer Science Department of the Federal University of Pernambuco, Recife PE, Brazil. She also works at GPRT a research group in the areas of computer networks and telecommunications. Currently she is involved in a number of research projects in the areas of network management, multimedia systems, the design of virtual reality systems and advanced communication devices.

Dênio Mariz received his Ph.D. in computer science from the Federal University of Pernambuco (Recife PE, Brazil) in 2005. He is currently an associate professor of computer networks at the Technical Federal Center (CEFET) in Paraiba, Brazil. His current research interests include common radio resource management algorithms,, and traffic measurement. He has been involved in several research projects funded by Brazilian agencies and also in partnership with telecom companies.

Djamel Sadok received his Ph.D. degree from Kent University in 1990. From 1990 to 1992, he was a Research Fellow in the Computer Science Department, University College London. He is currently a Professor of computer networks at the Computer Science Department of the Federal University of Pernambuco, Recife PE, Brazil. He is one of the cofounders of GPRT a research group in the areas of computer networks and telecommunications. His current research interests include traffic engineering of IP networks, wireless communications, broadband access, and network management. Dr. Sadok is a senior member of the IEEE Communications Society and currently leads a number of research projects with many telecommunication companies.