Unified Summaries for Internet traffic

Cristian Estan

May 12, 2004

Abstract

Traffic analysis is important to the operation of IP networks. The input to the analysis is raw data such as packet header traces or NetFlow records and the output is often the size aggregates such as the traffic generated by various applications or by individual customers. Storing the raw data allows the flexibility of running arbitrary new analyses in the future, but the sheer amount of raw data is often a challenge. Sampling based techniques such as smart sampling aim at reducing the amount of raw data while preserving the ability of future analyses to accurately estimate the traffic of any large aggregate.

There are three important measures of the traffic of an aggregate: the number of bytes, the number of packets and the number of flows. Current data reduction solutions allow estimating only one of these measures. In this paper we propose the idea of unified summaries that allow the analyses to get unbiased estimates for all three measures. Our unified summary that takes as input flow records is based on smart sampling and the one that reads in packet header traces is based on sample and hold. The most important contributions of this paper are the development of novel unbiased statistical estimators for the number of flows, the development of methods for combining summaries measuring bytes and packets using less memory than separate summaries, and experimental evaluation of the proposed solutions based on traces of traffic.

1 Introduction

Internet traffic consists of individual conversations called flows breaken into individual messages called packets. Often analyses aggregate the traffic into groups based on the fields defining the flows (e.g. web traffic, traffic coming from UCSD), and groups with large traffic are of most interest. The traffic of a group (also called aggregate) can be measured in packets or flows, and the two measures reveal different types of important information. Keeping track of all packets is impractical, and the traffic must be summarized. It is important to develop methods for compactly summarizing the traffic that allow low error approximate analyses. There are separate solutions for generating these summaries for the case when traffic is measured in packets and for the case when it is measured in flows. In this writeup I propose a solution that supports accurate unbiased analyses for both cases.

2 Summarization methods

In this paper I discuss three abstract summarization methods: ordinary sampling [7, 3], sample and hold [6, 4] and smart sampling [2]. Each of these computes in a single pass over the packet headers a traffic summary consisting of flow records. Each of these methods has as a tuning knob that allows the user to trade off the size of the summary (the number of records) and the accuracy of the analyses one can perform based on them. For simplicity I will assume for now that the flow records only count the number of packets. In Section 6 I will discuss counting bytes. The abstract summarization methods discussed here

compute a summary for a sequence of packets within a certain time interval and ignore information of the timing or ordering of the packets within the interval (the binned model [3]).

2.1 Ordinary sampling

As the stream of packets is arriving, packets are sampled independently at random with probability p. For the sampled packets the entry corresponing to the flow identifier in the packet is looked up in a hash table and the packet counter associated with that entry incremented. If there is no entry for the flow the packet belongs to, one is created and its packet counter initialized to 1. The flow identifier and the packet counter form the flow record.

2.2 Sample and hold

As the stream of packets is arriving, packets are sampled independently at random with probability p. For the sampled packets, unless the flow the packet belongs to already has an entry in the flow table, an entry is created and the packet counter initialized to 0. For all packets that have an entry, the entry's packet counter is incremented by 1. I effect this means that for each flow, its packets are sampled independently at random with probability p and after one packet is sammpled all packets belonging to the flow (including the sampled one) are counted.

2.3 Smart sampling

As the stream of packets is arriving, we build a hash table with a record for each active flow. At the end of the interval we have the exact number of packets belonging to each flow and the flow records are sampled independently at random with probabilities depending on their packet count. Flow records with one packet are sampled with probability p. Flow records with s packets are sampled with probability $\min(sp, 1)$. The packet counts in the sampled records are left unchanged.

Algorithm	Memory	Processing
Ordinary sampling	low	low
Sample and hold	low	high
Smart sampling	high	high

Table 1: Comparison of the resource consumption of the three algorithms

2.4 Estimating the traffic of aggregates

The user of the summary is interested in estimating the size of various aggregates containing one or more flows. The size can be measured in packets, flows or bytes. When one estimates the size of an aggregate, one first finds all flows in the summary that are part of the aggregate and sums up the individual estimates for each of these flows. By the linearity of expectation, the expected value for the estimate of the size of the aggregate is the sum of the expected values for the flow in the aggregate. By the independence of the sampling decisions for the flows, the variance in the estimate for the aggregate is the sum of the variances for individual flows. For the rest of this paper I will focus on the expectation and variance for the individual flows.

3 Brief comparison of summarization methods

While the three summarization methods are quite different, they are part of the same family of algorithms based on sampling and hash tables with per flow entries. For all three summarization algorithms, if we set $p = 1$, we obtain the list of all active flows with their exact packet counts. For all three algorithms, if the traffic mix contains only single packet flows, the output is a random sample of the traffic in the form of flow records with packet count of 1. But the less extreme configurations are more useful and for these, there are differences among the algorithms.

The processing and memory costs of these algorithms vary as shown in Table 1. Ordinary sampling

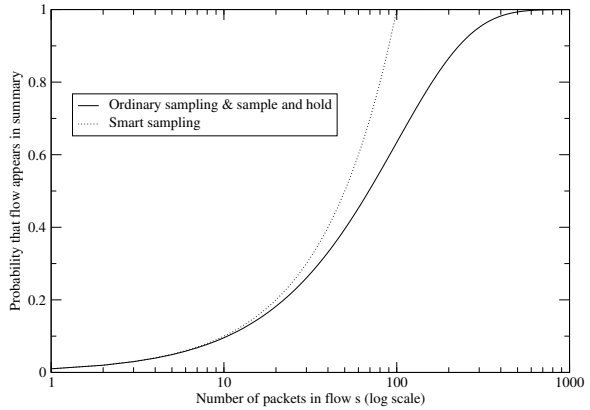

Figure 1: The probability for a flow getting an entry increases with the flow size. If all algorithms use the same sampling rate p, the probability for the creation of the entry is consistently higher for smart sampling, but it is never higher by more than 58.2% (which is achieved for flows of size $s = 1/p$).

uses little memory since it creates entries only for the sampled packets and performs little processing since it does table lookups only for the sampled packets. Sample and hold consumes just as little memory as ordinary sampling since it also creates entries only for the sampled packets, but it incurs high processing costs since it performs a lookup for each packet and updates the counter of the corresponding flow if found. Smart sampling uses high amounts of memory as it creates an entry for each flow. In the following sections we will see that the algorithms with higher resource consumption also have more accurate results.

When comparing the accuracy of the summaries produced by algorithms, it is fair to compare summaries of the same size, since for each algorithm, larger summaries give more accurate results. The size of the summaries depends on the packet sampling probability p. The expected size of the summary is the sum of the probabilities to have an entry for all active flows. Figure 1 shows the probability that a flow of size s has an entry in the summary, which is $1 - (1 - p)^s$ for ordinary sampling and sample and hold and $\min(ps, 1)$ for smart sampling. The entry creation probability is consistently higher for smart sampling, but but it is never higher by more than $1/(e - 1) = 58.2\%$ which is achieved for flows of size $s = 1/p$. Thus in the following sections I will compare the three algorithms using the same sampling probability. Ensuring that the expected report sizes were the same would require knowledge of the distribution of flows sizes. While this introduces a small bias towards smart sampling (because it will generate slightly larger summaries), it simplifies the comparison, since no knowledge of the distribution of flow sizes is needed.

4 Estimating packet counts

For ordinary sampling, the packet counter c of a flow of size s has a binomial distribution with parameters p and n. The case when $c = 0$ corresponds to the entry for the flow not being present in the summary. The unbiased packet count estimate for a flow record with c packets is $1/p \cdot c$. The variance of this estimate is $s/p(1 - p)$.

For sample and hold, the unbiased estimate for the number of packets is $c + 1/p - 1$. The variance of this estimate is $1/p(1/p-1)(1-(1-p)^s)$. See Appendix A for the details of the analysis.

For smart sampling the unbiased estimate for the number of packets is $\max(1/p, c)$. Its variance is $s \max(1/p - s, 0)$ [2].

In Figure 2 I compare the relative error (defined as the ratio of the standard deviation of the estimate for the flow size and the actual flow size) of the three algorithms for a packet sampling probability of $p = 1/100$ as the flow size s increases from 1 to 1,000. For algorithms, the relative error decreases as the size of the flow increases. Smart sampling's error drops to 0 when the flow size reaches $1/p$, since from there on, we get an exact flow count in the summary. Sample and hold's error is better than that for ordinary sampling and the difference becomes more and more pronounced as the flow size grows beyond $1/p$. For $s = 1/p$, ordinary sampling's erros is larger than sample and holds by $1/\sqrt{1 - e^{-1}} - 1 = 25.8\%$. Perhaps a more meaningful thing is to look at how

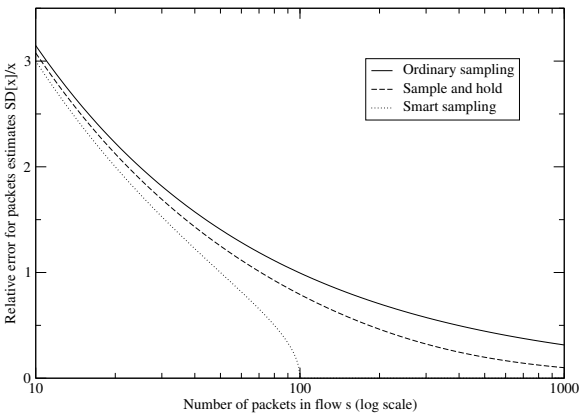

Figure 2: Comparison of the relative error of the packet estimates as a function of flow size for the three types of summaries for $p = 1/100$. The relative error is computed as the ratio between the standard deviation of the estimator and the actual flow size.

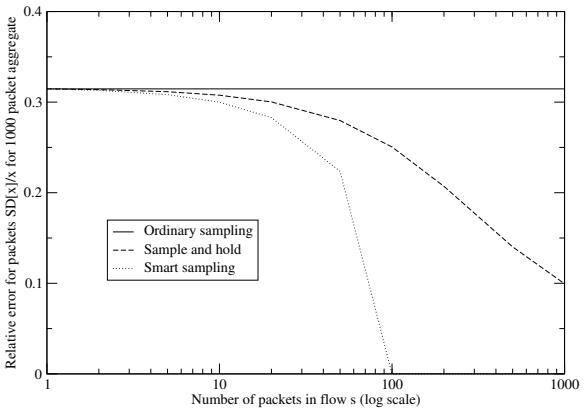

Figure 3: The relative error for an aggregate of 1000 packets decreases as the size of the flows increases if we use sample and hold or smart sampling, while it stays constant if we use ordinary sampling.

the relative error in the estimate of an aggregate of size 1,000 packets changes as the size of the flows that make it up increases from 1 to 1,000 packets. This is what Figure 3 shows. We can see how the error for ordinary sampling is not affected by the flow size while it decreases as the flow sizes go up for sample and hold and smart sampling.

5 Estimating flow counts

It has been proven [1] that one cannot get unbiased estimates for the number of flows from a random sample of the packets. Therefore we cannot estimate the number of flows from based on the ordinary sampling summary.

There is an estimator for the number of flows that is based on the sample and hold summary, and it is an original contribution of this paper. To estimate the number of flows in an aggregate, one finds all the matching flow records and counts the flows that have a packet counter larger than 1 as 1 flow and those that have a packet counter of exactly one packet as $1/p$ flows. It is obvious that this works for flows with 1 packet. It is plausible that it works for flows with $s >> 1/p$. Can we prove that it is an unbiased estimator for all flow sizes? With a probability α that depends on the size of the flow, one of the packets before the last one gets sampled. In this case, the contribution to the flow count of the estimate is 1, so the estimator is unbiased. With probability $1 - \alpha$ none of the packets before the last one gets sampled. Within this case, with probability p the last packet is sampled and our flow contributes $1/p$ to the total and with probability $1 - p$ it is not sampled and it contributes 0, thus the expectation is $p \cdot 1/p + (1 - p) \cdot 0 = 1$. The variance of this unbiased estimator is $(1 - p)^{s-1}(1/p - 1)$ as shown in Appendix A.

For smart sampling, the estimate for the number of flows is simple. Since the flow record has the exact size of the flow s, we know the exact probability that the flow got selected: $\min(1, ps)$. Therefore, we count each flow in the summary as $1/\min(1, ps)$ flows for the estimate. The variance of this estimator is $max(0, 1/(ps) - 1)$.

Figure 4 compares the relative error of sample and

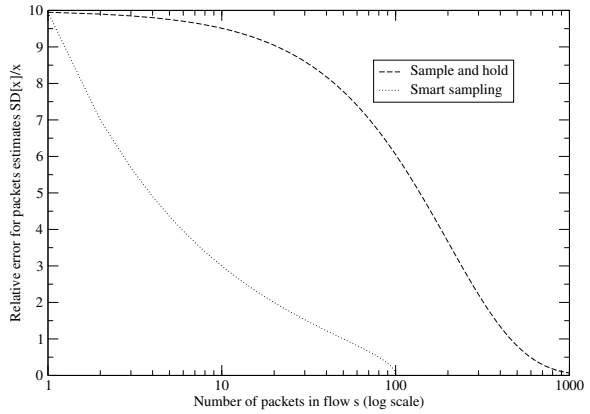

Figure 4: Smart sampling has lower error for flow count estimates that sample and hold.

hold and smart sampling for the flow count for a one flow aggregate as the flow size increases. While the sample and hold estimator is unbiased, it is clearly less accurate that that based on smart sampling for flows shorter than $1/p = 100$.

6 Estimating byte counts

With ordinary sampling we can add a byte counter to the flow records. Multiplying the value of this byte counter by $1/p$ gives an unbiased estimator for the number of bytes in the flow. If there is no limit on maximum packet size, the variance of this estimator can be unbounded. If packets have size at most b_{max} bytes, the variance of the estimate for the number of bytes of a flow with s packets is at most $b_{max}^2 s/p(1-p)$. The flows with large packets would have higher variance for their byte counts than the flows with small packets.

For sample and hold we can also add a byte counter to the entry, but there is no unbiased estimator for the number of bytes in a flow unless we make assumptions about the sizes of the packets in the flow (e.g. all packets have the same size). Another solution is to run a separate instance of sample and hold that samples bytes with probability q. One can estimate the number of bytes in a flow by adding $1/q - 1$ to the counter and the variance analysis from the packet case also carries over. The only problem is that instead of having one summary, we will have two: one for packet counts and one for byte counts.

For smart sampling, we can also count the number of bytes in the flow records. Since we know the exact sampling probability for the packet ps, multiplying by its inverse gives us an unbiased estimate for the number of bytes. However, just like with ordinary sampling, the variance of this estimate depends on size of the packets of the flow: flows with smaller packets will have lower variance than flows with larger packets. Like for sample and hold, we can run two instances for the algorithm, one tht provides unbiased low variance estimates for packet counts and one for bytes.

6.1 Sharing between byte and packet summaries

By correlating the choice of which record we sample for the packet and byte summaries, we can reduce the total size of the summary without affecting the accuracy of either of them.

7 Adding multiple summaries

An important question is computing the sum of multiple summaries. For example we have separate summaries for the hours of the day and we want to compute a summary for the whole day.

8 Conclusions

References

[1] S. Chaudhuri, R. Motwani, and V. Narasayya. Random sampling for histogram construction: How much is enough? In *Proceedings of the ACM SIGMOD*, 1998.

[2] Nick Duffield, Carsten Lund, and Mikkel Thorup. Charging from sampled network usage. In *SIGCOMM Internet Measurement Workshop*, November 2001.

[3] Cristian Estan, Ken Keys, David Moore, and George Varghese. Building a better netflow. In *Proceedings of the ACM SIGCOMM*, August 2004.

[4] Cristian Estan and George Varghese. New directions in traffic measurement and accounting. In *Proceedings of the ACM SIGCOMM*, August 2002.

[5] Philippe Flajolet. On adaptive sampling. In *Computing*, volume 34, pages 391–400. 1990.

[6] Phillip B. Gibbons and Yossi Matias. New sampling-based summary statistics for improving approximate query answers. In *Proceedings of the ACM SIGMOD*, pages 331–342, June 1998.

[7] Sampled netflow. `http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120s/120s11/12s_sanf.htm`.

A Analysis details

I focus the analysis on one flow of size s packets. Since our estimate for the size of an aggregate (whether traffic is measured in packets or flows) is the sum of the contribution of individual flows, by the linearity of expectation we can obtain the expected size of the aggregate by adding the expected contributions of the individual flows. Since the packet sampling decisions are independent, the variance in the estimate for the aggregate is the sum of the variances of the contributions of individual flows. Let $i < s$ be the number of packets missed before an entry for our flow is created and $p_i = p(1-p)^i$ be the probability to miss exactly i packets.

Let x be the number of packets our flow contributes to an aggregate. What is its expected value? We will prove by induction that $E[x] = s$.

Base case If s=1, the packet is sampled with probability p and in that case it is counted as $1+1/p-1 = 1/p$ packets. With probability $1-p$ it is not sampled (and it counts as 0). Thus $E[x] = p \cdot 1/p + 0 = 1 = s$.

Inductive step By induction hypothesis we know that for $s' = s-1$, $E[x'] = s' = s-1$.

$$\begin{aligned} E[x] &= p \cdot (s + 1/p - 1) + (1-p)E[x'] \\ &= ps + 1 - p + (1-p)(s-1) \\ &= ps + 1 - p + s - ps - 1 + p = s \end{aligned}$$

Let y be the number of flows our flow contributes to an aggregate (it can be 0, 1, or $1/p$) . What is its expected value? We can compute it by looking separately at the case where one packet before the last is sampled and at the case where no packet before tha last one is sampled (and this last case hgas two subcases based on whether the last packet is sampled or no). Let $\alpha = \sum_{i=0}^{s-2} p_i = 1 - (1-p)^{s-1}$ be the probability of catching one of the packets before the last one.

$$E[y] = \alpha \cdot 1 + (1-\alpha)(p \cdot 1/p + (1-p)0) = 1$$

What are the variances of x and y?

$$E[x^2] = \frac{1}{p}\left(\frac{1}{p} - 1\right)(1 - (1-p)^s) + s^2$$

Base case If s=1, $E[x^2] = p(1/p)^2 + 0 = 1/p$. Also $1/p(1/p-1)(1-(1-p)^1) + 1^2 = 1/p - 1 + 1 = 1/p$.

Inductive step By induction hypothesis we know that for $s' = s-1$, $E[x'^2] = 1/p(1/p-1)(1-(1-p)^{s'}) + s'^2 = 1/p(1/p-1)(1-(1-p)^{s-1}) + (s-1)^2$.

$$
\begin{aligned}
E[x^2] &= p\left(\frac{1}{p}-1+s\right)^2+(1-p)E[x'^2] \\
&= \frac{1}{p}+2(s-1)+p(s-1^2)+(1-p)(s-1)^2 \\
&\quad +(1-p)\frac{1}{p}(\frac{1}{p}-1)\left(1-(1-p)^{s-1}\right) \\
&= \frac{1}{p}-1+1+2(s-1)+(s-1^2) \\
&\quad +(1-p)\frac{1}{p}(\frac{1}{p}-1)-\frac{1}{p}(\frac{1}{p}-1)(1-p)^s \\
&= \frac{1}{p}(\frac{1}{p}-1)(p+1-p)+s^2 \\
&\quad -\frac{1}{p}(\frac{1}{p}-1)(1-p)^s \\
&= \frac{1}{p}\left(\frac{1}{p}-1\right)(1-(1-p)^s)+s^2
\end{aligned}
$$

$$
VAR[x]=E[x^2]-E[x]^2=\frac{1}{p}\left(\frac{1}{p}-1\right)(1-(1-p)^s)
$$

$$
\begin{aligned}
E[y^2] &= \alpha\cdot 1+(1-\alpha)\left(p\,(1/p)^2+(1-p)\cdot 0\right) \\
&= \alpha+(1-\alpha)1/p \\
VAR[y] &= E[y^2]-E[y]^2=\alpha+(1-\alpha)1/p-1 \\
&= (1-\alpha)\left(\frac{1}{p}-1\right)=(1-p)^{s-1}\left(\frac{1}{p}-1\right)
\end{aligned}
$$