Techniques for flow inversion on sampled data

Richard G. Clegg, Raul Landa, Hamed Haddadi and Miguel Rio
Dept of Electronic & Electrical Engineering
University College London
Corresponding author email: richard@richardclegg.org

Andrew W. Moore
Computer Laboratory
University of Cambridge

***Abstract*—The distribution of flow sizes is a quantity of interest fundamental to traffic engineering and network modelling and only likely to become more important in the future. The recovery of the flow-length distribution from (sampled) packet data is referred to as flow-inversion. Traditional packet sampling methods cause distortions in a recovered distribution of flow-length. We propose an improved method for inverting data sampled using the technique known as sample-and-hold. We show that the technique improves upon existing inversion techniques illustrated using both real and artificial data sets. The technique described may have applications to other inversion problems.**

I. Introduction

Sampling is a critical part of today's network measurement and monitoring. The volume of data traversing core routers makes it practically impossible for them to keep track of all the packets and their sources and destinations. Hence nearly all commercial routers nowadays implement sampling the packets and forming flow records, with the dominant format being that of Cisco NetFlow [1].

Although sampling eases the measurement and monitoring burden on core routers, it also lends itself to inaccuracies. Many smaller flows are missed and the longer flows can be truncated due to various time-outs. Much has been written on the problem of estimating statistics for flows, eg [1]–[3] and many others. The inversion problem is of critical importance for network operators. Aside from flow length being an obvious part of auditing and accounting, flow length estimation and inversion in general provides an effective mechanism to improve the accuracy of traffic-matrix computation [4]. In day-to-day network operations, the increasing adoption of streaming media and peer-to-peer applications makes it vital for an operator to be able to keep track of the larger flows, identifying heavy-hitters on the network and developing appropriate traffic shaping strategies in order to ensure adherence with the quality of service agreement levels.

It is common for network administrators to investigate the performance of a network by collecting sampled information about packets. The sampling method known as *sample-and-hold* is a method for sampling which is aimed at better estimates of long flows [1], [5]. This paper describes an inversion method for packet data sampled using sample-and-hold and tests it on real and artificial data sets.

A. Background and related work

A *flow* in a network is a set of packets which have the same 5–tuple (source IP address, destination IP address, source port, destination port and protocol). The *flow length distribution* is the set of probabilities that randomly selected flows have given lengths. Assume that for a given sample of packets there is some maximum flow length M (this may not be known) and therefore the distribution is $\{\theta_1, \ldots, \theta_M\}$. where θ_i is the probability that a randomly selected flow is of length i. The *flow inversion problem* is the problem of estimating the flow length distribution from sampled packet data.

One common sampling scheme is to sample every Nth packet. A similar sampling scheme is to sample in an independent and identically distributed (iid) manner (that is simply sampling each packet with a given probability p). The differences between these two methods can be important [6]. Duffield et al [2] used a Maximum Likelihood Estimator (MLE) based method for flow inversion on both schemes but encountered problems with adjusting the process to get accurate results. Hohn and Veitch [3] discuss inversion methods for iid sampling and come up with mathematically sound solutions although these have some practical limitations. Ribeiro et al [7] use several methods to estimate the flow distribution from iid sampled packets. Using features of the TCP protocol (sequence numbers and the SYN flag) they give

[1] http://www.cisco.com/univercd/cc/td/doc/cisintwk/intsolns/netflsol/nfwhite.htm

an MLE for flow lengths but only for "short" flows (in their paper, less than one hundred packet flows).

The majority of sampling techniques distort the flow distribution and are subject to one or more of the following problems: short flows may be totally missed; it is hard to estimate the length of long flows; flows may be misranked [8] and large flows may be split due to flow expiry [2]. Recent work on the flow inversion problem includes [2], [3], [7]. Previous researchers have noted that different sampling techniques may be desirable to improve the ability to recover longer flows [1], [5]. One such sampling method found in the literature is sample-and-hold. This method has advantages for the flow inversion problem. Cohen et al [9], [10] have produced an inversion method to recover the flow-length distribution from data sampled using sample-and-hold. The authors of this paper independently derived their method but improved upon its accuracy. In addition [9], [10] detail many ways to get useful statistical properties from data sampled using the sample-and-hold technique (and variants thereof).

The sample-and-hold method involves *tracked flows*. For each tracked flow its 5–tuple, as described in Section I, is stored. Every packet which is in the set of tracked flows is sampled. If a packet is not in the tracked set then this flow may be added to the set of tracked flows with a fixed probability $p \in (0, 1)$. (Note that to prevent the number of tracked flows growing until it consumes all available memory some method is needed to expire old flows. For a summary of some flow expiry issues see [11].)

The proportion of packets sampled for a given p is

$$P_{\text{samp}}(p) = 1 - \frac{1 - p - \sum_{i=1}^{M} \theta_i (1-p)^{i+1}}{p \sum_{k=1}^{M} k\theta_k},$$

where θ_i and M are as defined in the previous section.

The original description of sample-and-hold [5] proposed a probability varying with packet length, $1-(1-p)^b$, where $p \in (0, 1)$ and b is the length of the packet in bytes (it can be thought of as considering sampling every byte with probability p).

II. Methodology

A. Inverting sample-and-hold

The basic flow inversion for sample and hold is now given. A similar solution was independently discovered [9, Lemma 6.1] although the derivation is different. For each packet not in the set of tracked flows there is a probability p that the flow will be added to the set of tracked flows. Define $q = 1 - p$.

Let ϕ_i be the probability that i packets are sampled in a randomly chosen flow (note $\phi_0 \neq 0$ – some flows may have no packets sampled). Now,

$$\phi_i = \begin{cases} \sum_{j=i}^{\infty} p q^{j-i} \theta_j & i > 0 \\ \sum_{j=0}^{\infty} q^j \theta_j & i = 0. \end{cases}$$

Let X_i, $i \in \mathbb{N}$ be the distribution of flow lengths observed. The expectation value for X_i is given by,

$$\begin{aligned} \mathrm{E}[X_i] &= \mathbb{P}[\text{Sample length = } i | \text{Sample length} > 0] \\ &= \frac{\phi_i}{\sum_{k=1}^{\infty} \phi_k} = \frac{\sum_{j=i}^{\infty} q^j \theta_j}{q^i \sum_{j=1}^{\infty} q^j \theta_j \sum_{k=1}^{j} q^{-k}}. \end{aligned}$$

Evaluating $\sum_{k=1}^{j} q^{-k}$ gives,

$$\mathrm{E}[X_i] = \frac{(1-q)\sum_{j=i}^{\infty} q^j \theta_j}{q^i \sum_{j=1}^{\infty} q^j (q^{-j} - 1)\theta_j} = \frac{(1-q)\sum_{j=i}^{\infty} q^j \theta_j}{q^i [1 - \sum_{j=1}^{\infty} q^j \theta_j]}. \quad (1)$$

Subtracting $q\mathrm{E}[X_{i+1}]$ from $\mathrm{E}[X_i]$ and rearranging gives the final answer $\theta_i = (\mathrm{E}[X_i] - q\mathrm{E}[X_{i+1}])/(1 - q + q\mathrm{E}[X_1])$.

This is an exact solution but $\mathrm{E}[X_i]$ is unknown. Obviously X_i is an unbiased estimator for $\mathrm{E}[X_i]$ and it can be seen that, therefore, an unbiased estimator for θ_i is

$$\hat{\theta}_i = \frac{X_i - qX_{i+1}}{1 - q + qX_1}. \quad (2)$$

This is similar to [9, Lemma 6.1]. Their version does not give the normalising constant $1/(1 - q + qX_1)$ but this could trivially be calculated since the θ_i must sum to one. Note that this equation is not guaranteed to be in the range $[0, 1]$. In particular negative values regularly occur when $X_{i+1} \gg X_i$. Obviously one could arbitrarily set negative values to zero but this would have two undesirable effects, firstly the estimator would no longer be unbiased and secondly the estimated distribution would then sum to more than one. Because these negative values are more likely to occur in the tail of the distribution, introducing a minimum of zero and rescaling the distribution would also produce a bias by increasing the probability of longer distributions.

B. Improving this inversion

If $\mathrm{E}[X_i]$ is known then the previous calculations would completely solve the problem. While X_i is an unbiased estimator for $\mathrm{E}[X_i]$ it may have a high coefficient of variance. In particular, when $\mathrm{E}[X_i]$ is small a problem occurs since X_i is the observed proportion of flows of length i then it must, by definition, be an integer divided by the total number of observed flows. Consider, for example, a sample with one thousand observed flows, then X_i can take values in $\{0, 0.001, 0.002, \ldots\}$. If the true value of $\mathrm{E}[X_i]$ is 0.00001 then X_i will not be a reasonable estimate. Since it is likely that nearby values

of $\mathrm{E}[X_i]$ are close for large i then $\hat{e}_i$, an improved estimator for $\mathrm{E}[X_i]$ for large i, might be given by a weighted sum of nearby values.

$$\hat{e}_i = \frac{\sum_{j=-n(i)}^{n(i)} w_j X_{j+i}}{\sum_{k=-n(i)}^{n(i)} w_k}, \tag{3}$$

where the w_j are a series of weights and $n(i)$ is a *window size* which depends on i. The question then is how to select w_j and also $n(i)$.

Firstly, the problem of picking the weights will be dealt with. A common assumption with flow distributions is that they have a heavy-tail. Assume initially that the flow length distribution is a Zeta distribution (this assumption will be weakened later to heavy-tailed and the consequences of the assumption not being met will be examined experimentally) $\theta_i = \zeta(\alpha) i^{-\alpha}$ for some $\alpha \in (1,3)$ where $\zeta(\alpha)$ is the Riemann–Zeta function. Assume that the data has been sampled using sample-and-hold with probability parameter p (and let $q = 1-p$ as usual). Therefore, substituting the above formula for θ_i for the zeta distribution into (1) gives

$$\begin{aligned}\mathrm{E}[X_i] &= \frac{(1-q)q^{-i}\sum_{j=i}^{\infty} q^j \zeta(\alpha) j^{-\alpha}}{1 - \sum_{j=1}^{\infty} q^j \zeta(\alpha) j^{-\alpha}} \\ &= C_{q,\alpha} q^{-i} \sum_{j=i}^{\infty} q^j j^{-\alpha}, \end{aligned} \tag{4}$$

where $C_{q,\alpha}$ is a constant fixed for a given q and α. It is given by

$$C_{q,\alpha} = \frac{\zeta(\alpha)(1-q)}{1 - \sum_{j=1}^{\infty} q^j \zeta(\alpha) j^{-\alpha}}.$$

From (4) for $i+1$ and $i-1$ then

$$\begin{aligned}\mathrm{E}[X_{i+1}] &= \mathrm{E}[X_i] q^{-1} - C_{q,\alpha} q^{-1} i^{-\alpha} \\ \mathrm{E}[X_i] &= q\mathrm{E}[X_{i+1}] + C_{q,\alpha} i^{-\alpha} \\ \mathrm{E}[X_{i-1}] &= q\mathrm{E}[X_i] + C_{q,\alpha}(i-1)^{-\alpha}\end{aligned}$$

Substitute to get

$$\begin{aligned}\mathrm{E}[X_i] &= q\mathrm{E}[X_{i+1}] + \left(\frac{i-1}{i}\right)^{\alpha} [\mathrm{E}[X_{i-1}] - q\mathrm{E}[X_i]] \\ &= q\mathrm{E}[X_{i+1}] + \left(1 + \sum_{k=1}^{\infty} \binom{\alpha}{k} (-i)^{-k}\right) \\ &\quad [\mathrm{E}[X_{i-1}] - q\mathrm{E}[X_i]] \\ &= \frac{q\mathrm{E}[X_{i+1}] + \mathrm{E}[X_{i-1}]}{1+q} + \\ &\quad \frac{\left(\sum_{k=1}^{\infty} \binom{\alpha}{k}(-i)^{-k}\right)[\mathrm{E}[X_{i-1}] - q\mathrm{E}[X_i]]}{1+q},\end{aligned}$$

where $\binom{\alpha}{k} = 1/k! \prod_{j=0}^{k-1}(\alpha - j)$. For $\alpha \in (1,3)$ then $|\binom{\alpha}{k}| < 2$ since $\binom{\alpha}{k} = [(\alpha-1)/1][(\alpha-2)/2]\cdots[(\alpha - k + 1)/k]$ and the modulus of each of the terms is less than 1 apart from the first which is at most 2. Since $\sum_{k=1}^{\infty}(-i)^{-k} = 1/(i-1)$ then the right hand term is $O(1/i)$. Therefore

$$\mathrm{E}[X_i] = \frac{q\mathrm{E}[X_{i+1}] + \mathrm{E}[X_{i-1}]}{1+q} + \varepsilon,$$

where $\varepsilon \sim O(1/i)$ is an error term and

$$|\varepsilon| \leq \frac{2(\mathrm{E}[X_{i-1}] - q\mathrm{E}[X_i])}{i-1},$$

hence a good approximation for large i is given by

$$\mathrm{E}[X_i] \simeq \frac{q\mathrm{E}[X_{i+1}] + \mathrm{E}[X_{i-1}]}{1+q}.$$

Similar manipulations will yield that for $k \ll i$,

$$\mathrm{E}[X_i] \simeq \frac{q^k \mathrm{E}[X_{i+k}] + \mathrm{E}[X_{i-k}]}{1+q^k}, \tag{5}$$

although the bounds on the error term grow weaker as k gets larger.

This leads to a possible scheme for choosing the weights w_j in (3),

$$w_j = \begin{cases} 1 & j = 0 \\ q^j(1 - j/[n(i)+1]) & n(i) \geq j > 0 \\ (1 + j/[n(i)+1]) & -n(i) \leq j < 0. \end{cases} \tag{6}$$

This includes a linear fall off which reduces the w_j to 0 outside the window $n(i)$ in addition to the q^j factor from (5). In fact this linear fall off makes no major difference and the results are largely unaffected without it.

An obvious question is how this is affected when the distribution is not a zeta distribution. For a heavy-tailed distribution where $\theta_i = K i^{-\alpha}$ for large i, some $K > 0$ and $\alpha \in (1,3)$ will yield exactly the same result. Many heavy-tailed distributions have this approximate form. The question of what happens if the distribution does not have a heavy-tail is dealt with empirically in section III-B.

C. The final estimation procedure

A final issue remaining is the choice of window size $n(i)$. The critical issue is how many sampled flows had a given size i packets. If the number of sampled flows of size i is high then X_i is likely to be a good estimate of $\mathrm{E}[X_i]$. So for $i = 1$ a window size of zero (which means simply $\hat{e}_i = X_i$) is likely to still get a reasonable estimate. On the other hand, for large i, in a given sample it is likely that there were no flows at all with size exactly i packets and the window size should be increased. However, if the window size is too large the error in (5) will also become large. One obvious strategy is to set a desired number of sampled flows within the window size. Let T to be the desired number of samples within the window. That is, the window size should be adjusted so that T or more flows were observed with packet lengths in the range $i - n(i)$ to $i + n(i)$. The estimation procedure then becomes the following.

1) Set $i := 1$ and the sample window used is $n := 1$.

2) Get an estimate for $\mathrm{E}[X_i]$ using the weights in (6) in conjunction with (3).
3) Use this to get an estimate for θ_i using (2).
4) If fewer than T flows were observed with packet lengths in the range $i-n$ to $i+n$ then increase the sample window $n := n+1$.
5) Set $i := i+1$. If i is less than the largest flow length available in the observed data then go to step 2.

Note that the last step terminates the algorithm when observations run out. This is practically necessary but does mean that the inverted distribution will, by necessity, not estimate the tail of the original distribution. For reasons of practicality, in these experiments, a maximum window size of 1,000 was enforced. This is because, in extreme cases with a few very fat flows of 100,000 packets the algorithm was having to estimate the flow size at hundreds of thousands of points using a window size of tens of thousands.

III. Results

The results on simulated and real data are shown in the following sections. The experiments are first performed on simulated data with a zeta distribution in Section III-A. Simulated data using a non-heavy tailed distribution is tried in Section III-B. Real data from several sources is tested in Section III-C.

In this section, the graphs are presented on a logscale as a complimentary cumulative distribution function (CCDF), $\mathbb{P}[X > x]$ versus x where x is a given flow length. In fact the data given here are troublesome to display in any form. Because of the nature of the estimation procedure, the estimated probabilities can be negative as noted in [9] and this remains true even for the improved estimates. The CCDF is no longer strictly non-increasing and can become negative hence some values cannot be seen on a logscale.

The errors in estimating the sample distribution are given by the following procedure. Let o_i be the value of the CCDF at point i before sampling. Let e_i be the estimated value of the CCDF at point i after inversion. Let l be the lowest flow length of interest and h be the highest flow length of interest. Two error measures are used here, the mean error (which is a measure of bias in the data)

$$\varepsilon_m(l,h) = \frac{\sum_{i=l}^{h} o_i - e_i}{h-l+1}$$

and the mean absolute error

$$\varepsilon_a(l,h) = \frac{\sum_{i=l}^{h} |o_i - e_i|}{h-l+1}.$$

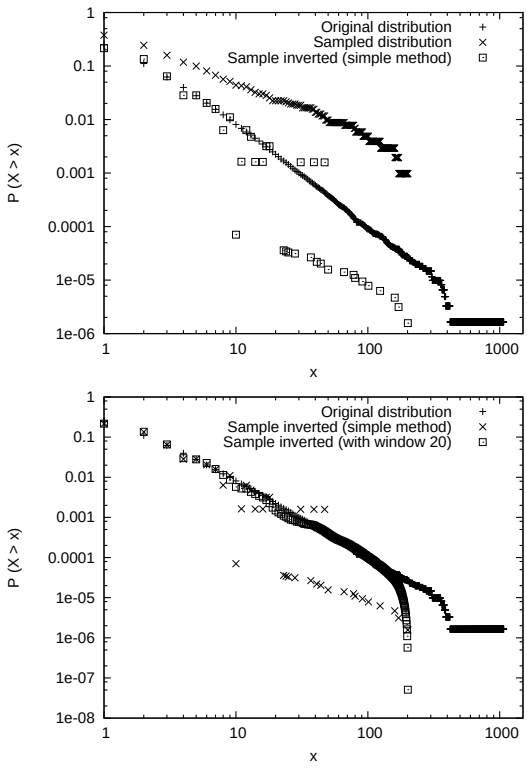

Fig. 1. The distribution of sampled and reconstructed flow lengths for packets where flow lengths have a zeta distribution.

The notation $\varepsilon_m(1,-)$ or $\varepsilon_a(1,-)$ will be used to indicate the error over all flow lengths from 1 to the maximum flow length present in the reconstructed sample (which is the maximum flow length in the sampled data).

A. Results on simulated data with a zeta distribution

The top part of figure 1 shows results using simulated data for a million packets. The flows in this experiment have a zeta distribution with $\alpha = 2$ and the simplest correction method using (1). The figure (and all figures in this section) plots $\mathbb{P}[X > x]$ vs x on a logscale. The most obvious thing from this plot is the severe distortion to the flow length distribution. As can be seen, the reconstruction is very good for $x < 5$, quite good for $x < 10$ but becomes very poor for $x > 20$.

The lower part of figure 1 shows the same data set reconstructed with the algorithm given in Section II-C with $T = 20$ and the windows set as in (6). As can be seen, the inversion is greatly improved when compared with Figure 1.

The top of Table I shows the errors as described in the introduction to this section using inversion with and without the window. The method called "Simple" is the

reconstruction just using the method of Section II-A. For methods using windows parameters from (6) the value of the parameter T is given. Window parameters $T = 1, 20, 100, 500$ are shown here. As will be seen the method is relatively insensitive to this parameter (a desirable property) and the value 500 is large enough that errors begin to increase again.

From the table first we can see that the results for the window method is, largely an improvement on the results using the simple method. The exceptions are the results where $T = 500$ and for $\varepsilon_m(1, -)$ which is slightly worsened. The reason for this may be that the simple estimator was already an unbiased estimator for the probability that a flow had a given length and hence the mean error might be expected to be low already. The method can be seen not to have great sensitivity to the value of T and, for example, the results for $T = 20$ and $T = 100$ do not vary greatly.

	$\varepsilon_m(1,20)$	$\varepsilon_a(1,20)$	$\varepsilon_m(1,-)$	$\varepsilon_a(1,-)$
Zeta distribution				
Simple	-0.0012	0.0041	0.0032	0.0069
$T = 1$	0.00028	0.0039	0.0047	0.0056
$T = 20$	0.00051	0.0027	0.0048	0.0054
$T = 100$	0.0027	0.0027	0.0051	0.0054
$T = 500$	0.0095	0.0095	0.006	0.0063
Normal distribution				
Simple	-0.18	0.3	-0.14	0.25
$T = 1$	-0.088	0.27	-0.056	0.22
$T = 20$	-0.082	0.17	-0.054	0.12
$T = 100$	-0.086	0.098	-0.052	0.08
$T = 500$	-0.12	0.12	-0.085	0.11

TABLE I
ERROR ANALYSIS FOR THE ESTIMATION ON THE ZETA DISTRIBUTION AND NORMAL DISTRIBUTION.

B. Results of simulated data which is not heavy-tailed

The next obvious test is to test on some simulated data which definitely does not meet the assumption of heavy-tailed flow lengths. In this section, therefore, a simulated data set ridiculously far from this assumption is created. The flow lengths of the data set in this section are chosen to have a normal distribution with mean 100 and variance 20. This is obviously a hopelessly unrealistic model for real data but should test whether the method used fails if the assumption of heavy tails is not met. Again a million packets are generated using this assumption and sampled using sample-hold with $p = 0.001$.

The bottom half of table I shows the errors calculated using inversion techniques on the normal distribution data. In this case, perhaps surprisingly, it can be seen that the window method is a great improvement although this data set does not meet the assumptions that the method was designed for. Again the sensitivity to the window parameter T is not great which is a positive sign. While the errors in the inversion remain high in this case, the window method much more than halves them in the best case $T = 100$.

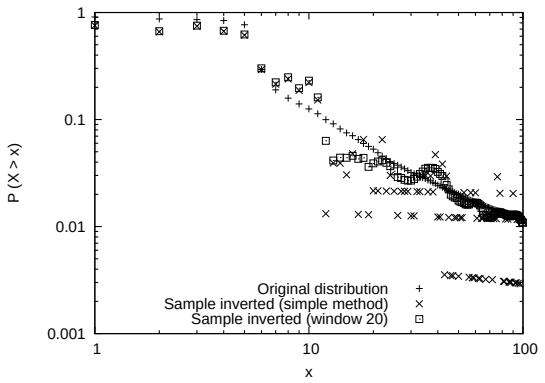

Fig. 2. Reconstruction of the QUAINT data using the simple method and a window with $T = 20$.

C. Results on real data

The same tests were performed on four real data sets, two from the CAIDA website [2], one from the QUAINT project [12] and one from the NLANR project [3]. For full details on the data consult the references given. The data set CAIDA 1 is 7.5 million packets and 5500 flows. The data set CAIDA 2 is 11 million packets and 7535 flows. The NLANR data is 47 million packets and 26000 flows. The QUAINT data is 2.7 million packets and 1200 flows. In all cases the methodology was the same. The data was processed into flows using no sampling to get the base case to compare with and then sample using sample-and-hold with $p = 0.001$ and inverted using the techniques from Section II. Figure 2 shows the CCDF reconstructed using inversion for the QUAINT data using the simple method and a windowed method with $T = 20$.

Table II shows the results for all the real data analysed. As can be seen, the windowed method provided improved reconstruction of the flow distribution, in many cases, greatly so. Because of the larger sample sizes, the errors were in general lower here than the artificial data tested. As would be hoped the method is robust to changes in window size and this is not a critical parameter.

IV. CONCLUSIONS AND FUTURE WORK

This paper has demonstrated a technique for recovering estimates for the flow length distribution from data sampled using the sample-and-hold technique. The

[2] http:www.caida.org

[3] http://pma.nlanr.net/Special/leip2.html

	$\varepsilon_m(1,20)$	$\varepsilon_a(1,20)$	$\varepsilon_m(1,-)$	$\varepsilon_a(1,-)$
CAIDA data set 1				
Simple	-0.014	0.016	-0.0016	0.0017
$T=1$	-0.012	0.015	1×10^{-5}	0.00013
$T=20$	-0.012	0.015	1.2×10^{-5}	9.8×10^{-5}
$T=100$	-0.011	0.012	1.6×10^{-5}	8.3×10^{-5}
CAIDA data set 2				
Simple	0.0045	0.0096	-0.00097	0.001
$T=1$	0.0055	0.0099	1.1×10^{-5}	2.3×10^{-5}
$T=20$	0.0055	0.0099	1.2×10^{-5}	1.9×10^{-5}
$T=100$	0.0053	0.0081	1.2×10^{-5}	1.6×10^{-5}
QUAINT data				
Simple	-0.038	0.067	-0.0087	0.0087
$T=1$	-0.029	0.065	-5.1×10^{-5}	7.1×10^{-5}
$T=20$	-0.027	0.061	-5×10^{-5}	6.6×10^{-5}
$T=100$	-0.022	0.059	-4.8×10^{-5}	6.4×10^{-5}
NLANR data				
Simple	-0.0076	0.0079	-0.00037	0.00037
$T=1$	-0.0073	0.0077	9.5×10^{-7}	2.3×10^{-5}
$T=20$	-0.0073	0.0077	1.1×10^{-6}	2.2×10^{-5}
$T=100$	-0.0073	0.0077	1.2×10^{-6}	2.2×10^{-5}

TABLE II
ERROR ANALYSIS FOR THE ESTIMATION ON REAL DATA.

simplest method used has been seen before in the literature. Inversion techniques involving averaging estimates over a window create an improved estimate. While these techniques were developed based on the assumption of heavy-tailed flow distribution, they remain valid even when this assumption is completely violated in the data. The improved inversion techniques work very well on four real data sets. Although the choice of window is somewhat ad hoc, the method is not sensitive to the specifics of this choice.

Further research remains to be done in this area. The techniques given here estimate flow lengths only up to the length of the largest flow available in the sampled data. This will miss the tail of the real (unsampled) distribution. The exact choice of window given here is somewhat ad hoc, however, tests show that the results are not sensitive to the window parameters used. While some optimisation might be done here, it seems that there may be diminishing returns in exactly optimising the choice of window. As mentioned in section II the CCDF is non-monotone. Mathematical techniques exist to produce a monotone function which is closest to a non-monotone function. The most straightforward ideas (negative probabilities set to zero and then normalised so the sum is one) would introduce systematic biases into the flow distribution.

The estimates could certainly be improved by other techniques. In particular, long flows are poorly estimated and flows longer than the maximum flow observed in sampling are not estimated. This is simply because the authors have no good method for estimating the maximum flow length in the original data and this, itself, would seem to be a research problem of some interest. One approach taken by some authors is to use features of the TCP protocol (for example SYN flags) to increase inversion accuracy. Another possibility would be to look at correlations in the TCP/IP header fields (source and destination port) since different types of traffic would be expected to have different flow length distributions.

Acknowledgments

This work was funded under the MASTS project, EPSRC grants GR/T10503/01, GR/T10510/03, and GR/T10527/01.

REFERENCES

[1] C. Estan and G. Varghese, "New directions in traffic measurement and accounting: Focusing on the elephants, ignoring the mice," *ACM Trans. Comput. Syst.*, vol. 21, no. 3, pp. 270–313, 2003.

[2] N. Duffield, C. Lund, and M. Thorup, "Estimating flow distributions from sampled flow statistics," *IEEE/ACM Trans. Netw.*, vol. 13, no. 5, pp. 933–946, 2005.

[3] N. Hohn and D. Veitch, "Inverting sampled traffic," in *IMC '03: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*. New York, NY, USA: ACM Press, 2003, pp. 222–233.

[4] Y. Zhang, M. Roughan, C. Lund, and D. L. Donoho, "An information-theoretic approach to traffic matrix estimation," in *SIGCOMM '03*, 2003, pp. 301–312.

[5] C. Estan and G. Varghese, "New directions in traffic measurement and accounting," in *SIGCOMM '02: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM Press, 2002, pp. 323–336.

[6] M. Roughan, "A comparison of Poisson and uniform sampling for active measurements," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 12, pp. 2299–2312, 2006.

[7] B. Ribeiro, D. Towsley, T. Ye, and J. Bolot, "Fisher information of sampled packets: an application to flow size estimation," in *IMC '06: Proceedings of the 6th ACM SIGCOMM on Internet measurement*. New York, NY, USA: ACM Press, 2006, pp. 15–26.

[8] C. Barakat, G. Iannaccone, and C. Diot, "Ranking flows from sampled traffic," in *CoNEXT'05: Proceedings of the 2005 ACM conference on Emerging network experiment and technology*. New York, NY, USA: ACM Press, 2005, pp. 188–199.

[9] E. Cohen, N. Duffield, H. Kaplan, C. Lund, and M. Thorup, "Sketching unaggregated data streams for subpopulation size queries," in *Proc. of the 2007 ACM Symp. on Principles of Database Systems (PODS 2007)*. ACM, 2007.

[10] ——, "Algorithms and estimators for accurate summarization of internet traffic," in *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, 2007.

[11] R. Sommer and A. Feldmann, "NetFlow information loss or win?" in *IMW '02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*. ACM Press, 2002, pp. 173–174.

[12] A. J. Field, U. Harder, and P. G. Harrison, "Measurement and modelling of self-similar traffic in computer networks," *IEE Proceedings – Communications*, vol. 151, no. 4, pp. 355–363, 2004.