Fisher Information of Sampled Packets: an Application to Flow Size Estimation *

Bruno Ribeiro Don Towsley
University of Massachusetts at Amherst
{ribeiro,towsley}@cs.umass.edu

Tao Ye Jean Bolot
Sprint Advanced Technology Laboratories
{Tao.Ye,Bolot}@sprint.com

UMass CS Technical Report 06-37
July 2006

Abstract

Packet sampling is widely used in network monitoring. Sampled packet streams are often used to determine flow-level statistics of network traffic. To date there is conflicting evidence on the quality of the resulting estimates. We take a systematic approach, using the Fisher information metric and the Cramér-Rao bound, to understand the contributions of different types of information within sampled packets on the quality of flow-level estimates. We apply this approach to the estimation of TCP flow size distributions, and the benefits of including SYN flag and TCP sequence number information on estimation error. Our approach predicts that sequence number information substantially reduces errors in estimating flow size distributions. Furthermore, additional SYN flag information significantly reduces estimation error for short flows. We present a Maximum Likelihood Estimator (MLE) that relies on all of this information and show that it is efficient, even when applied to a small sample set. We validate our results with Tier-1 Internet backbone traces and evaluate the benefits of sampling from multiple monitors. Our results show that combining estimates from several monitors is 50% less accurate than an estimate based on all samples.

1 Introduction

Data reduction is an indispensable component of today's Internet measurement and monitoring. With the increase in network utilization, it is very difficult for monitoring applications to process every packet in the aggregated backbone links at OC48$^+$ levels. Recently, many data streaming algorithms have focused on summarizing network traffic with a very small memory footprint [18], [12], often beneficial to inline monitoring at the router. While lightweight, this aggregation requires prior knowledge of interested statistics before it can be implemented at the monitoring point. On the other hand, sampling methods require very little inline computation, but transmit a subset of traffic to a powerful backend server for analysis. This allows users both flexibility and extensibility in deploying measurement and monitoring applications at the server. Sampling also helps reduce the processing load, and memory and storage demands of monitoring

*This material is based upon work supported by the National Science Foundation under Grant No. ITR 0325868. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

systems. However, some information content is inherently lost with sampling. This work presents a theoretical framework within which to assess how much information of a given flow level metric remains after sampling. While we primarily focus on the estimation of the flow size distribution, our framework applies to other metrics as well. Moreover, we quantify the value of TCP header fields for the estimation of flow size distributions.

Many sampling schemes have been proposed, from general purpose packet sampling and flow sampling, to methods aimed at identifying traffic elephants such as smart sampling [5] and sample-and-hold [6]. Two standardization efforts, PSAMP [21] and IPFIX [20], are current underway as well. Among these, random or periodic (close to random) packet sampling, (sFlow [22]), flow summarization of packet level information (Cisco NetFlow [19]), and a combination of both (Cisco sampled NetFlow) are popular methods deployed in commercial networks. Random packet sampling consists of independently selecting each packet for processing with probability p. Periodic sampling is shown to have similar characteristics as random sampling [3]. While packet sampling generally provides detailed and accurate packet level characteristics, it is not clear whether it can reveal detailed flow level characteristics.

The flow size distribution is an important metric that has received some attention in recent years. Flow size is the number of packets in a flow. We are interested in estimating the flow size distribution, i.e. the fraction of flows that contains i packets during a measurement interval, with i typically small. This is an important metric for many applications, such as traffic engineering, and denial of service attack and worm/virus outbreak detections. It has been previously thought to be very difficult to estimate the flow size distribution accurately from sampled traffic [9]. In the first work in the field, Duffield et al [3] provided several estimators, but did not provide a proof of their accuracy.

In this work we use the Fisher information metric to address many open questions concerning flow size distribution estimation from packet sampling. This is possible because of the tie between Fisher information and estimation mean squared error through the Cramér Rao lower bound. Using the Fisher information, we identify certain TCP fields that contain high informational value beneficial to flow size estimation. We also explore the benefits of computing estimates at a central site when samples come from multiple monitors. We will also observe that our framework simplifies the task of analyzing and developing estimation algorithms for sampling at both a single monitor and at multiple monitors. Products of our study are estimators that comes close to optimal, even when given a small number of samples. We validate our results using traces taken from a Tier-1 backbone network. We focus on TCP flows as they account for 80-90% of packets in the network [23]. However, our approach is quite general and can be applied to other network estimation problems.

The rest of the paper is organized as follows: In Section 2 we introduce the general model of obtaining flow-level statistics under a random packet sampling scheme. Then we lay out the information theory framework and compute the Fisher Information in Section 3. The development of an efficient estimator that achieves the Cramér Rao bound, Maximum Likelihood Estimator (MLE), follows in Section 4. We evaluate using real traces in Section 5, and evaluate the benefit of multiple monitors in Section 6. Finally we conclude with Section 7.

2 A Model for Packet Sampled Flows

We introduce a model of flow sizes and sampled flows produced through packet sampling. We first define the relevant entities and then enhance the model to include SYN and sequence number information.

The conventional IP flow definition is a set of packets that obey the following rules:

- Any two packets have the same 5-tuple, i.e., the same IP Source, IP Destination, source port number, destination port number, and protocol number.
- Maximum inter-packet arrival time must be less than a threshold t, where t is a value given by the network operator, typically between 30 to 60 seconds.

In practice, some systems use other protocol information such as a FIN packet in TCP to terminate a flow. Cisco NetFlow evicts flows that are active for more than time t, typically 30min, to free memory for new flows. Here we choose the conventional definition to keep our model straightforward.

We monitor packets at a chosen point in the network. At this monitor, packets are sampled according to a Bernoulli process with sampling probability p, $0 < p < 1$. We refer to the flows prior to sampling as *original flows*. A sampled (or thinned) flow is a flow that has at least one of its packets sampled. A flow of size i is a flow that originally has i packets. Likewise, a sampled flow of size m is a flow that has m packets sampled, where $m \geq 1$. Some original flows are not sampled and therefore not observed. Some original flows may split into multiple sampled flows. We do not account for flow splitting.

2.1 Basic Model

Assume the original flow size is upper bounded by $W > 0$. Let θ_i be the fraction of original flows of size i that crossed the monitor during some given time interval and let θ'_i be the fraction of original flows of size i that were sampled. Under the Bernoulli sampling process assumption: $\theta'_i = (1-(1-p)^i)\theta_i$. Let $\vec{\theta} = (\theta_1, \ldots, \theta_W)^{\mathrm{T}}$ denote the original flow size distribution. Likewise, let $\vec{\theta'} = (\theta'_1, \ldots, \theta'_W)^{\mathrm{T}}$ denote the distribution of the sampled flow sizes. $\vec{\theta}$ and $\vec{\theta'}$ are related as follows:

$$\theta_i = g_i(\vec{\theta'}) = \frac{\theta'_i/(1-(1-p)^i)}{\sum_{k=1}^{W}[\theta'_k/(1-(1-p)^k)]}. \tag{1}$$

Our objective is to estimate $\vec{\theta}$ from the sampled flows. Note that $\vec{\theta}$ is constrained by $\sum_{i=1}^{W}\theta_i = 1$ and $0 \leq \theta_i \leq 1$, $\forall i$. These constraints also apply to $\vec{\theta'}$.

Let $\mathcal{L}$ be a set of label tuples. A label $j \in \mathcal{L}$ can be, for instance, the number of packets obtained in a sampled flow. Let $j \in \mathcal{L}$ be a label given to a sampled flow and let d_j be the fraction of sampled flows with label j. For now consider j to be the number of packets obtained from a sampled flow and let $\vec{d} = (d_1, \ldots, d_W)$ denote the sampled flow size distribution. Distributions $\vec{d}$ and $\vec{\theta}$ are related by

$$d_j = \sum_{i=1}^{W} b_{i,j}\theta_i, \tag{2}$$

where $b_{j,i}$ is the binomial probability of sampling j packets out of i original packets given sampling rate p.

Equation (2) can be written in vector notation as

$$\vec{d} = \mathbf{B}\vec{\theta}, \tag{3}$$

where $\mathbf{B}$ is a $W \times W$ matrix whose element (i,j) is $b_{j,i}$. Matrix $\mathbf{B}$ is an upper triangular matrix and thus (3) has a unique solution. A similar relationship also holds for $\vec{\theta'}$.

Let $\hat{D}_j$, $j \geq 0$, denote the total number of flows with j sampled packets and $n = \sum_{\forall k} \hat{D}_k$ the number of sampled flows. We can also further define $\hat{\vec{d}}^{(n)}$ as

$$\hat{d}_j^{(n)} = \hat{D}_j^{(n)}/n. \tag{4}$$

In what follows we omit the dependence of $\hat{\vec{d}}$ on n for notational convenience.

In the next section we extend the model to account for SYN and sequence number information.

2.2 TCP SYN and sequence numbers

The basic model only accounts for the number of packets inside a sampled TCP flow. A sampled flow can carry more information about its original IP flow size, through stateful upper layer protocols. TCP [15], in particular, has two fields that provide further information regarding length: control flags and sequence numbers.

Flow samples with SYN packets. As pointed out in [3], the TCP SYN flag provides valuable information during the estimation phase. As in [3], we assume original flows include exactly one SYN packet, which is the first packet of the flow. We denote a sampled flow starting with a SYN packet as a SYN sampled flow. Because there is only one SYN packet per flow, the distribution $\vec{\theta}'$ conditioned on the SYN sampled flows is the same as the original flow size distribution $\vec{\theta}$. We refer to a TCP sampled flow with a SYN sampled packet as TCP SYN sampled flow. We use $\hat{d}_{(\mathrm{S},k)}$ to denote the fraction of the sampled flows where a SYN packet is sampled and there are k sampled packets. Likewise, we denote by $\hat{d}_{(\mathrm{N},k)}$ the fraction of sampled flows where there was no SYN sampled packet and there are k sampled packets in total. For now we focus on TCP SYN sampled flows, and ignore flows without a sampled SYN. In Section 4.3 we will see how to add flows sampled without SYNs the estimation. Equation (3) holds for only TCP SYN sampled flows with $\mathbf{B}$ properly redefined. The modification to $\mathbf{B}$ is found in [3]. We refer an estimator that uses $\hat{d}_{(\mathrm{S},k)}$ as a "SYN-pktct" estimator.

Next we turn our attention to sequence numbers.

Samples with TCP sequence numbers. TCP uses a 32-bit sequence number that counts bytes in a flow. Assume that the TCP start sequence number, the starting byte count of the sampled packets, is available. An estimator that measures flow sizes in number of bytes can clearly benefit from TCP sequence numbers. The question is whether an estimator based on packet counts can also benefit from sequence numbers. We assume that there is a function $h(s_a, s_b)$ that takes two TCP sequence numbers s_a and s_b from two distinct packets a and b form the same flow and returns the number of packets sent between a and b including a and b. For now we will not worry over the fact that it is hard to find the exact packet count using TCP sequence numbers. In Section 5 we will provide a reasonably good approximation to h.

Let $s_{min}^{(u)}$, $s_{max}^{(u)}$ be the smallest/greatest sampled TCP sequence number values of flow u (wrap around is easily treated). Let U be a set of sampled SYN flows and let $\hat{d}_r$ be the fraction of sampled SYN flows with $r = h(s_{max}^{(u)}, s_{min}^{(u)})$, $\forall u \in U$. This new sample definition induces a new matrix $\mathbf{B}$ with $b_{i,(\mathrm{S},r)} = p(1-p)^{i-r}$, $\forall i \leq W, 2 \leq k \leq i$ and $b_{i,(\mathrm{S},1)} = (1-p)^{i-1}$, and the rest of the matrix being zero. Denote an estimator that uses $\hat{d}_{(\mathrm{S},k)}$ as a "SYN-seq" estimator.

We have introduced several types of information. One question that remains is which type of information can be considered valuable to our estimator. The next section is devoted to quantifying the impact of these types of information on the estimation accuracy of $\vec{\theta}$.

3 Fisher Information in flow size estimation

This section quantifies the improvement on estimation achieved by adding different types of information to the sampled flow distribution $\vec{d}$. Throughout this work we focus on unbiased estimators. Let θ_i denote the quantity to be estimated and $T(\theta_i)$ its estimate. An unbiased estimate guarantees $E[T(\theta_i)] = \theta_i$. In what follows we consider only unbiased estimators, unless stated otherwise.

Let $T(\vec{\theta})$ be an estimate of $\vec{\theta}$ obtained by an estimator T. A good unbiased estimator of a flow size i is characterized by a low mean squared error $E[(\theta_i - T(\theta_i))^2]$. This motivates the definition of an *efficient estimator*.

Efficient estimator: An estimator T of θ_i is said to be *efficient* if its mean squared error, $E[(\theta_i - T(\theta_i))^2]$, is the minimum among all estimators.

In what follows we provide way to compute a tight lower bound on the mean squared error for flow size unbiased estimators.

3.1 Measuring information: Fisher information matrix

The Fisher information matrix of a single sampled flow can be thought of as the amount of information that the observable random vector $\vec{d}$ carries about the unobservable parameters $\vec{\theta}$ or $\vec{\theta'}$ upon which the probability distribution of $\vec{d}$ depends. The results in this section derived for $\vec{\theta}$ are also valid for $\vec{\theta'}$.

The Fisher information can be defined over a set of samples. If the samples in the set are all mutually independent, then the Fisher information of the set of samples is the sum of the Fisher informations of each of the samples [2].

Assume that flows take at least two sizes and $\theta_i > 0, 1 \leq i \leq W$.

Let $n\hat{d}_j$ denote the number of sampled flows with label j as defined in Section 2.1. Assume $n = 1$ and that our sole sampled flow has sample label j'. Note that in this scenario $\hat{d}_{j'}^{(1)} = 1$ and zero otherwise. Define an operator $(\cdot)_k$ over a vector that retrieves its k-th element. Let

$$\alpha(\hat{\vec{d}}^{(1)}; \vec{\theta}) = \sum_{\forall k} \hat{d}_k^{(1)} (\mathbf{B}\vec{\theta})_k = \sum_{\forall k} \hat{d}_k^{(1)} d_k \tag{5}$$

be the conditional probability that our sole sampled flow has sample label j' given flow size distribution $\vec{\theta}$. α is also known as the likelihood function. The likelihood function α can be extended to $\alpha^{(n)}$, the likelihood of n independently sampled flows. The parameters $\vec{\theta}$ of the likelihood function $\alpha^{(n)}$ are constrained by:

$$\sum_{\forall i} \theta_i = 1 \tag{6}$$

and

$$0 < \theta_i < 1, \ \forall i. \tag{7}$$

The constraints (7) can be included in α by a simple change in variables

$$\theta_i = \beta(\gamma_i) = \frac{1}{1 + \exp(-\gamma_i)}, \tag{8}$$

with $\gamma_i \in \mathbb{R}$. Function β maps γ_i with domain $\mathbb{R}$ to $(0, 1)$, thus satisfying constraints (7). Furthermore, define a function $g(\vec{\gamma}) = \sum_{\forall i} \beta(\gamma_i) - 1$. Then $g(\vec{\gamma}) = 0$ iff constraint (6) is satisfied. Take $\vec{\gamma} \in \mathcal{D}$, where

$\mathcal{D} = \{\vec{\gamma} | g(\vec{\gamma}) = 0\}$ and $\beta(\vec{\gamma})$, $(\beta^{-1}(\vec{\gamma}))$, a vector whose i-th element is $\beta(\gamma_i)$, $(\beta^{-1}(\gamma_i))$, then the likelihood function f of one sampled flow is

$$f(\hat{\vec{d}}^{(1)}; \vec{\gamma}) = \alpha(\hat{\vec{d}}^{(1)}; \beta^{-1}(\vec{\gamma})).$$

Under the above conditions we find the Fisher information matrix of the flow size estimation problem. Let $\nabla_{\vec{\gamma}} \ln f(\hat{\vec{d}}^{(1)}; \vec{\gamma})$ be a vector whose i-th element is $\partial \ln f(\hat{\vec{d}}^{(1)}; \vec{\gamma}) / \partial \gamma_i$. We use the main result of [8] to find the pseudo-inverse of the Fisher information matrix with equality constraints on its parameters. Let $P(\hat{d}_j^{(1)} = 1) = d_j$ be the probability that our sole sampled flow has sample label j and

$$\mathbf{J}(\vec{\gamma}) = \sum_{\forall j} (\nabla_{\vec{\gamma}} \ln f(\hat{\vec{d}}^{(1)}; \vec{\gamma})) \ (\nabla_{\vec{\gamma}} \ln f(\hat{\vec{d}}^{(1)}; \vec{\gamma}))^{\mathrm{T}} \, d_j, \tag{9}$$

also with

$$\mathbf{G}(\vec{\gamma}) = \nabla_{\vec{\gamma}} g(\vec{\gamma}). \tag{10}$$

From now on we omit the dependence of $\mathbf{J}$ and $\mathbf{G}$ on $\vec{\gamma}$ for notational convenience. Let $\mathcal{I}$ be the Fisher information matrix of $f(\vec{d}^{(1)}; \vec{\gamma})$. We obtain $\mathcal{I}$ from its pseudo-inverse $\mathcal{I}^{-1}$. The pseudo-inverse of the Fisher information matrix with $\vec{\gamma} \in \mathcal{D}$, $\mathcal{I}^{-1}(\vec{\gamma})$, is a $W \times W$ matrix

$$\mathcal{I}^{-1}(\vec{\gamma}) = \mathbf{J}^{-1} - \mathbf{J}^{-1}\mathbf{G}^{\mathrm{T}}(\mathbf{G}\mathbf{J}^{-1}\mathbf{G}^{\mathrm{T}})^{-1}\mathbf{G}\mathbf{J}^{-1}, \tag{11}$$

where $\mathbf{G}^{\mathrm{T}}$ is the transpose of $\mathbf{G}$. Note that the Fisher information depends on the original flow size distribution $\vec{\theta}$. For notational convenience we omit the dependence of $\mathcal{I}$ on $\vec{\theta}$.

The Fisher information can be used to compute a lower bound on the error of any unbiased estimator of $\vec{\theta}$ as seen next.

3.2 The Cramér-Rao bound

The Cramér-Rao theorem states that the mean squared error of any unbiased estimator is lower bounded by the inverse (or pseudo-inverse) of the Fisher information matrix [8], provided some regularity conditions on the log-likelihood function f are satisfied.

Lemma 3.1 *Let $\mathcal{I}$ be the Fisher information matrix of one sampled flow. If packets are sampled independently according to a Bernoulli process (as in Section 2), the Fisher Information matrix of n sampled flows is $n\mathcal{I}$.*

Proof If packets are sampled independently and according to a Bernoulli process, then flows are also sampled independently. The Fisher information of a set of n independently sampled flows is $n\mathcal{I}$ [2]. □

The regularity conditions required by the Cramér-Rao bound as given in [10] translates to $\forall i \sum_j \partial d_j / \partial \gamma_i = \partial / \partial \gamma_i \sum_j d_j$ on the flow size estimation problem, which clearly holds.

Let $\tilde{\gamma}_i$ be an unbiased estimate of γ_i. Combining the Cramér-Rao theorem with Lemma 3.1 gives

$$E[(\gamma_i - \tilde{\gamma}_i)^2] \geq -(\mathcal{I}^{-1})_{i,i}/n,$$

or, more generally

$$E[(\vec{\gamma} - \tilde{\vec{\gamma}})(\vec{\gamma} - \tilde{\vec{\gamma}})^{\mathrm{T}}] \geq -\mathcal{I}^{-1}/n, \tag{12}$$

with $\mathcal{I}^{-1}$ from equation (11).

The mean squared error obtained from (12) is a function of parameters $\vec{\gamma}$. We would like to find the mean square error with respect to $\vec{\theta}$.

The mean squared error of $\vec{\theta}$ follows by applying the delta method [16]: Let n be a large number of sampled flows. Altough n is assumed to be a large number, it can still be considered small on the scale of a Tier-1 Internet backbone. Let $\mathbf{H} = [h_{i,j}]$ with $h_{i,j} = \beta(\gamma_j)/\partial\gamma_i$ and likewise $\mathbf{H'} = [h'_{i,j}]$ where $h'_{i,j} = \partial g_i(\beta(\vec{\gamma}))/\partial\gamma_j$, with g_i, i, j as defined by (1). Thus in the case where the original likelihood function α is a function of $\vec{\theta}$, the mean squared error of the estimate of $\tilde{\vec{\theta}}$ is

$$E[(\vec{\theta} - \tilde{\vec{\theta}})(\vec{\theta} - \tilde{\vec{\theta}})^{\mathrm{T}}] \geq \mathbf{H}(-\mathcal{I}^{-1}/n)\,\mathbf{H}^{\mathrm{T}} \tag{13}$$

and when α is a function of $\vec{\theta'}$,

$$E[(\vec{\theta} - \tilde{\vec{\theta}})(\vec{\theta} - \tilde{\vec{\theta}})^{\mathrm{T}}] \geq \mathbf{H'}(-\mathcal{I}^{-1}/n)\,\mathbf{H'}^{\mathrm{T}} \tag{14}$$

3.3 Applying the Cramér-Rao bound

We illustrate the application of the Cramér-Rao bound with two examples. The first one in Section 3.3.1 shows all of the necessary steps to obtain the Cramér-Rao bound. The second one in Section3.3.2 displays the use of the Fisher Information through the Cramér-Rao bound, in designing better estimators.

3.3.1 Example with maximum flow size of two

Let $W = 2$ be the maximum flow size. Let $\theta_1 = 0.88$ and $p = 0.01$. From equation (10), we have $(\mathbf{G})_i = \theta_i^2/(\theta_i - 1)$. Equation (9) yields $\mathbf{J}(\beta^{-1}(\theta_1)) = -\vec{e}_1\,\vec{e}_1^{\mathrm{T}}/d_1 - \vec{e}_2\,\vec{e}_2^{\mathrm{T}}/d_2$, where $\vec{e}_j = (b_{j,1}, b_{j,2}) \cdot (\vec{\theta}^2/(\vec{\theta} - 1))$. Let j denote the number of sampled packets in a SYN sampled flow. Then $b_{1,1} = 1$, $b_{2,1} = 0$, $b_{1,2} = 0.99$ and $b_{2,2} = 0.01$. The pseudo-inverse of the Fisher information $\mathcal{I}^{-1}$ (equation (11)) of one sampled flow is

$$\mathcal{I}^{-1} = \begin{bmatrix} -1078 & 1078 \\ 1078 & -1078 \end{bmatrix}$$

Now assume n flows are sampled. Thus the lower bound on the mean squared error of estimates $\tilde{\gamma_1}$ and $\tilde{\gamma}_2$ obtained using the Cramér-Rao bound will be $E[(\gamma_1 - \tilde{\gamma}_1)^2] \geq 1078/n$ and $E[(\gamma_2 - \tilde{\gamma}_2)^2] \geq 1078/n$. The Cramér-Rao bound of parameters $\vec{\theta}$ comes from the delta method. Matrix $\mathbf{H}$ is

$$\mathbf{H} \approx \begin{bmatrix} 0.105 & 0 \\ 0 & 0.105 \end{bmatrix}.$$

Thus from (13), the mean squared error of any unbiased estimates $\tilde{\theta}_1$ and $\tilde{\theta}_2$ of θ_1 and θ_2 respectively are: $E[(\theta_1 - \tilde{\theta}_1)^2] \geq 1092/n$ and $E[(\theta_2 - \tilde{\theta}_2)^2] \geq 1092/n$ for n sampled flows given n sufficiently large.

3.3.2 Valuable information from TCP sequence numbers

Consider the problem of estimating flow size distribution using packet counts, and SYN and sequence number information as defined in Section 2.2. The elements of $\mathbf{B}$ are $b_{i,1} = (1-p)^{i-1}$ and $b_{i,j} = p(1-p)^{i-j}$ for $j > 1$. For our next example, assume a maximum flow size $W = 4$ and $\vec{\theta} = (0.56, 0.08, 0.18, 0.18)$.

We compute the Cramér-Rao bound for a sampling rate of $p = 1/10$. Also consider the estimation using packet counts and SYN information (SYN-pktct) as defined in Section 2.2. Figure 1 shows the Cramér-Rao bound under this scenario. Clearly the addition of TCP sequence numbers drastically increases the Fisher information of the samples. This increase in the Fisher information is translated into a much smaller lower bound on estimation error. As W increases, the difference in the errors of these two estimations increases until the bars of the confidence interval of the SYN-pktct estimator becomes greater or equal to 1, while the SYN-seq estimator increases its error slightly as we will see in Sections 4.2 and 5.

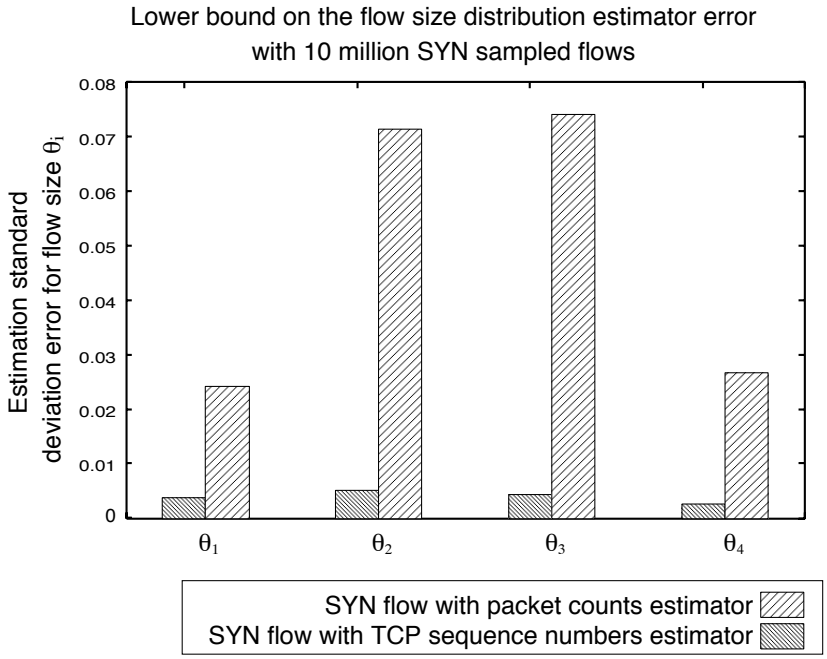

Figure 1: Cramér-Rao bounds of the examples on Section 3.3.2. This graph compares the estimation of SYN-pktct to the SYN-seq. Notice that adding TCP sequence numbers to the estimation greatly improves its quality.

We will shortly present MLEs for the SYN-pktct, SYN-seq estimators. Experimentally we will find that these MLEs are efficient in that they approach the Cramér-Rao bound even for a small sample size, $n = 10,000$. We show that it is also efficient even when sampled flows without SYN packets are included.

4 Finding a good estimator

The Maximum log-Likelihood Estimator (MLE), finds a set of parameters $\tilde{\vec{\theta}}$ that maximize the log-likelihood of the sampled data. Under the same regularity conditions as required for the Cramér-Rao bound, the MLE is an asymptotically *efficient* unbiased estimator of $\vec{\gamma}$, i.e., its error achieves the Cramér-Rao lower bound as the number of samples tends to infinity. As in practice we do not have a very large number of samples, we would like it to be *efficient* using the number of samples typically collected at Tier-1 backbone routers. This section presents a MLE for the SYN-seq model. That does not require a large number of samples to achieve the Cramér-Rao error lower bound. In addition, we present a Conjugate Gradients algorithm for the MLE, a faster convergence algorithm than the commonly used Expectation Maximization algorithm.

We estimate the MLE over function $\alpha^{(n)}$ through the use of penalty functions for the constraints in (6) and (7). Whenever a value of $\vec{\theta}$ violates one of the constraints, the likelihood function receives a penalty, which in

the end forces the search to remain within the constrained region. To simplify analysis, we generate synthetic sampled flows for the traffic in an idealized fashion. In this section we estimate the flow size distribution using only SYN sampled flows. This, of course, does not account for packet sampling introduced "noise" such as flow-splitting, which counts one long original flow into two or more shorter ones. However [11] describes how to treat that flow splitting. We will evaluate the complete model with "noise" in Section5 on actual traces. Next we introduce the MLE for our model.

4.1 MLE with Conjugate Gradients

Let n be the number of sampled flows and $n\hat{d}_k$ the number of sampled flows with label k. The likelihood function with respect to parameters $\vec{\theta}$, as defined in Section 3.1, is $\alpha^{(n)}(\hat{\vec{d}};\tilde{\vec{\theta}})$. The MLE can be written as

$$\tilde{\vec{\theta}} = \arg\max_{\tilde{\vec{\theta}}} \; n \prod_{k=1}^{\mathrm{W}} \hat{d}_k \; \ln(\mathbf{B}\tilde{\vec{\theta}})_k \tag{15}$$

subject to $\sum_i \tilde{\theta}_i = 1$ and $0 < \tilde{\theta}_i < 1, \; \forall i \in \{1, \ldots, W\}$.

First we consider the SYN-pktct MLE as proposed in [3]. We analyze the Expectation Maximization (EM) algorithm, used in [3] to find a solution of the log-likelihood equation (15). Let $\hat{D}_{(\mathrm{S},r)}$ denote the number of SYN sampled flows with label r sampled packets. Let $\hat{d}_{(\mathrm{S},r)}$ be the fraction of SYN sampled flows with r sampled packets, as defined by (4).

We detail the approach in [3] for the sake of completeness. The EM algorithm finds the MLE $\tilde{\vec{\theta}}$ by successive refinement of previous estimates:

$$\tilde{\theta}_i^{(k+1)} = \tilde{\theta}_i^{(k)} \sum_{i \geq r \geq 1} \frac{b_{i,r}\hat{d}_{(\mathrm{S},r)}}{\sum_{k>r} \tilde{\theta}_k^{(k)} b_{k,r}},$$

where $\tilde{\vec{\theta}}^{(0)}$ is an initial guess of $\vec{\theta}$.

Although the EM algorithm is sound, needs no fine tuning, and is guaranteed to always improve the estimate at each step, in practice it can suffer from slow convergence [14]. More specifically, Theorem 5.2 in [14] shows that if the parameters $\vec{\theta}$ are "poorly separable" then EM exhibits a slow convergence rate. The term "poorly separable" can be quantified as the difficulty of distinguishing whether a sample j came from flow sizes i or i' with $i \neq i'$, i.e., if $b_{i,j}\theta_i \approx b_{i',j}\theta_{i'}$. Unfortunately, flow size estimation suffers from this vileness. Although one expects that other maximum likelihood algorithms will also suffer with these "poorly separable" parameters, it is believed that in practice the effect is be felt more by EM [14] (conjecture strengthened by our practical experience with our EM and Conjugate Gradients method implementations when applied to our problem).

We instead use the method of Conjugate Gradients [13] to compute a solution to (15). Our Conjugate Gradients C routine was implemented with the help of the wnlib library[1].

For the above algorithm to work, we only need to provide the matrix $\mathbf{B}$ and the gradient $\nabla_{\tilde{\vec{\theta}}} \ln \alpha^{(n)}(\hat{\vec{d}};\tilde{\vec{\theta}})$ conditioned on $\sum_{i=1}^{W} \theta_i = 1$ and $0 < \tilde{\theta}_i < 1, \; \forall i \in \{1, \ldots, W\}$. The i th component of our gradient is

$$\frac{\partial}{\partial \theta_i} \ln \alpha^{(n)}(\hat{\vec{d}};\tilde{\vec{\theta}}) = \sum_{j=1}^{i} \frac{b_{i,j}\, \hat{d}_j}{\sum_{k \geq j} \tilde{\theta}_k \, b_{k,j}} - 1.$$

[1] http://www.willnaylor.com/wnlib.html

All constraints are introduced as penalty functions. Like EM, the Conjugate Gradient algorithm also requires an initial guess $\tilde{\vec{\theta}}^{(0)}$. The only requirement for any initial guess is that no flow size have zero probability.

4.2 SYN-seq MLE: the use of TCP sequence numbers

The MLE is an asymptotically unbiased estimator of $\vec{\theta}$, i.e., $E[\tilde{\vec{\theta}}] = \vec{\theta}$ as $n \to \infty$. From the application of the data processing theorem on the Fisher information matrix [17], the best estimator of flow sizes that uses TCP sequence numbers will perform better, or at least no worse, than an estimator that does not take sequence numbers into account. This is expected as one can always throw the sequence number information away inside the estimator.

Let $W = 50$ be the maximum flow size and $p = 1/10$ be the packet sampling rate. In the following experiments we use samples from a renormalized flow size distribution obtained from the Sprint backbone network. The flow size distribution renormalization creates a distribution that is a re-scaled true Internet flow size distribution but with maximum flow size W. The original distribution came from trace BB-East-1, summarized at the beginning of Section 5 in Table 1.

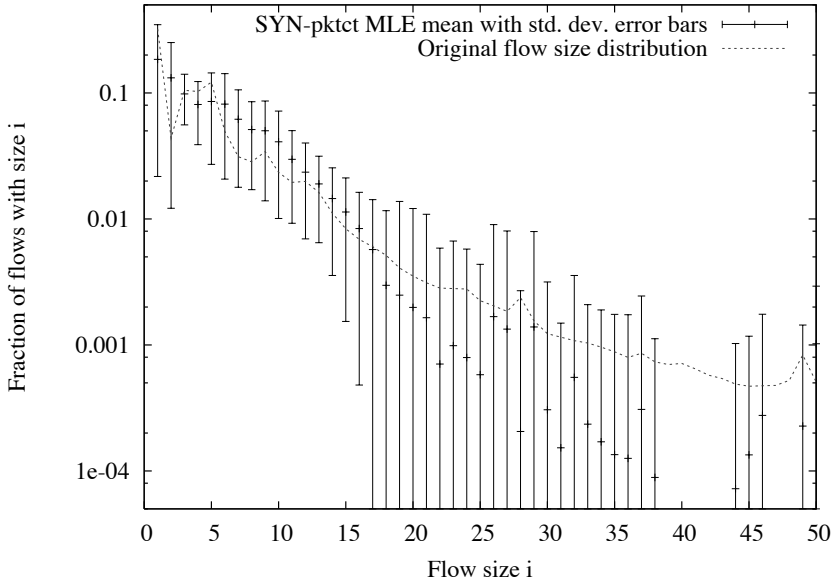

Figure 2: Flow size distribution obtained from SYN-pktct MLE from SYN sampled flows (no TCP sequence number). Observe the large standard deviation errors of the estimates (from 150 independent trials with 100,000 SYN sampled flows each and packet sampling probability $p = 1/10$).

The following experiment uses 100,000 synthetically generated SYN sampled flows. The first estimation is done using the SYN-pktct MLE. In order to assess the performance of this estimator,$\hat{\theta}_i^{(0)} = 1/W$ is used as an initial value for the optimization.

Figure 2 shows a graph of the original flow size distribution and estimation using the SYN-pktct MLE. Observe the large standard deviation errors. Large standard deviation errors of the estimates are not the worst part, as the estimation is also biased, i.e., even averaging the results of 150 independent trials, $E[\hat{\theta}_i] \neq \theta_i$ for most flow sizes i, but particularly bad for small flow sizes. This is due to the initial values chosen for$\tilde{\vec{\theta}}^{(0)}$ as shown in Section 4.4. It is also an indication that the estimator is not accurate with only 100,000 sampled flows.

SYN-seq MLE, the estimator that includes TCP sequence number information, reduces estimation errors

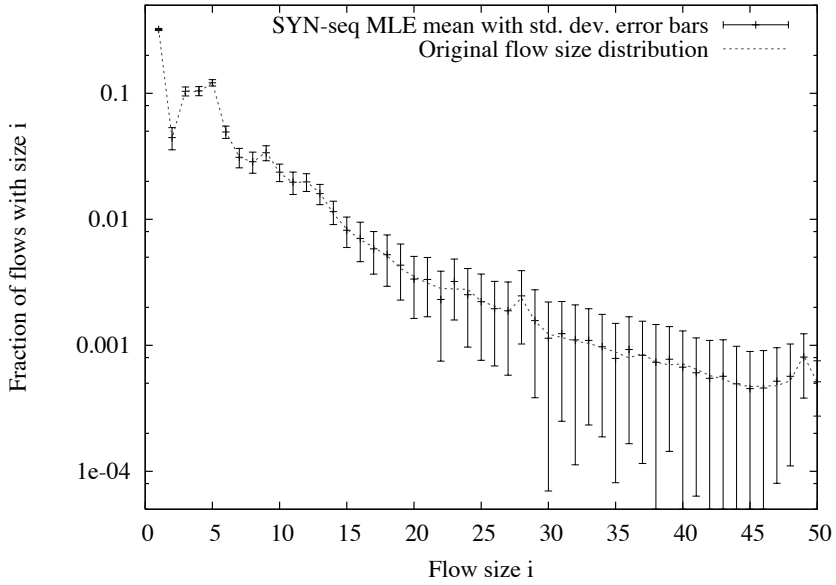

Figure 3: Flow size distribution obtained from SYN-seq MLE from 100,000 SYN sampled flows. Observer the small standard deviation errors of the estimates (from 100 independent trials with 100,000 SYN sampled flows each and packet sampling probability $p = 1/10$).

significantly, as shown in Figure 3. Observe the small standard deviation errors of the estimates when TCP sequence numbers are added to the estimator. The mean of the estimator is the true mean as one expects from a good unbiased estimator ($E[\hat{\theta}_i] = \theta_i\,,\ \forall i \in \{1, \ldots, N\}$). The additional information has a huge impact on reducing the estimator variance.

In what follows we show that the MLE for SYN sampled flows with TCP sequence number of the scenario described above is *efficient* even with small sample sets.

The MLE is guaranteed to achieve the Cramér-Rao lower error bound asymptotically.

Figure 4 shows the error of SYN-seq MLE in estimating flow sizes compared to its respective Cramér-Rao bound. For a large number of samples ($\geq 10^6$) the Cramér-Rao bound and the SYN-seq MLE mean squared error are indistinguishable. Thus the Cramér-Rao bound is tight and the SYN-seq MLE appears to be *efficient* even when there are at least $10,000$ samples. Increasing the maximum flow size W also increases slightly the minimum required number of samples. Our experiments shows that when $W = 200$ $100,000$ sampled flows are required to achieve the bound. For $W = 300$, the bound is still achieved with $100,000$, which we conjecture remains true even for larger flow sizes.

In practice the Cramér-Rao bound has another use: One can define the desired values of the standard deviation errors of the estimates and obtain the minimum number of samples necessary to achieve it.

Although SYN sampled flows have high information content, in practice packet sampling rates are commonly as low as $p = 1/100$ in the backbone network, significantly reducing the number of SYN sampled flows. With such a low sampling rate one must use all sampled flows, not only those with SYN packets. In what follows, sampled flows with missing SYN packet are taken into account in the estimation.

4.3 Adding all sampled flows to the estimator

In our traces, TCP flows account for a sizable fraction of the flows. About 20% of the TCP sampled flows in our traces contain a SYN sampled packet. In [3] the authors conjecture that there are usually not enough

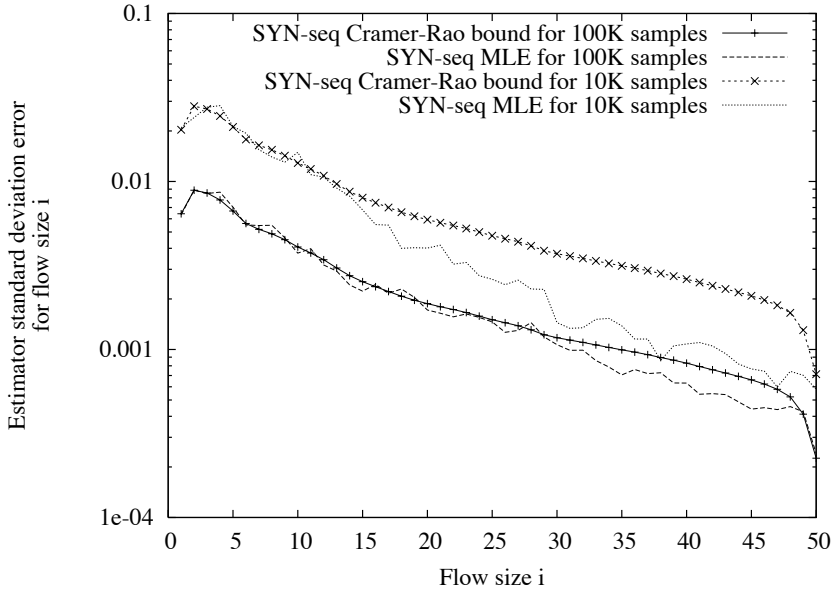

Figure 4: This graph shows the standard deviation errors of the estimates. Here we compare the SYN-seq MLE against the Cramér-Rao bound using 100,000 and 10,000 SYN sampled flows. We can see that the SYN-seq MLE achieves the bound even for a small number of sampled flows.

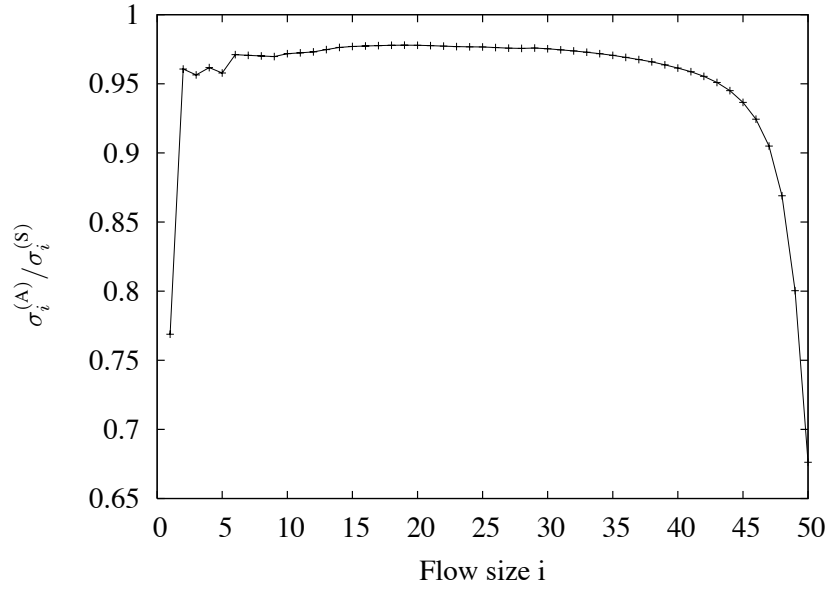

Figure 5: The graph of $\sigma_i^{(\mathrm{A})}/\sigma_i^{(\mathrm{S})}$ shows that including all sampled flows with no sampled SYN packet on the estimator is beneficial for the θ_1 estimate $\tilde{\theta}_1$. It also shows that it has little impact on other flow sizes.

SYN sampled flows for accurate flow size distribution estimation with their estimator.

The data processing theorem [17] states that adding information can only increase the Fisher information. Therefore, it is possible to increase the precision of the SYN-seq estimator by incorporating all sampled flows with no SYN packets. Incorporating these samples can be done seamlessly in the SYN-seq estimator and even in the SYN-pktct estimator. This extension can potentially increase the accuracy of the maximum likelihood estimators presented in [3]. However we will restrict our analysis to the TCP sequence number estimator. Denote the estimator that uses TCP sequence numbers and SYN flags (SYNs and non SYN flows) as "ALL-seq-sflag estimator". Let **B** denote the sampling probability matrix as defined by (3). Let j denote a tuple $(\text{SYNFLAG}, r)$, where $r = h(s^{(u)}_{max}, s^{(u)}_{min})$ with h as defined in Section 3.3.2. Let SYNFLAG = S when there is a SYN packet in the sampled flow and SYNFLAG = N otherwise. Thus $b'_{i,<\text{S},k>} = p\,(1-p)^{i-k}$ and $b'_{i,<\text{N},k>} = (i-k)\,p\,(1-p)^{i-k}$. The element i, j of matrix **B** is $b_{i,j} = b'_{i,j} / \sum_{\forall j} b'_{i,j}$.

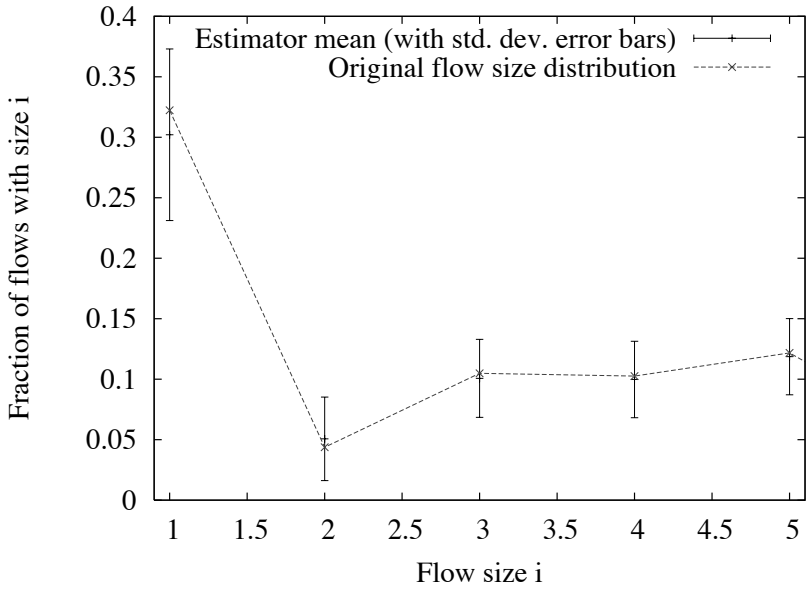

Figure 6: SYN-seq MLE mean with standard deviation error bars.

Consider the impact of this new information on the Cramér-Rao bound. Once again we generate synthetic samples from the rescaled flow size distribution of the **BB-East-1** trace given $W = 50$ as the maximum flow size and $p = 1/200$ as the packet sampling rate. In this scenario 19% of the TCP sampled flows have a SYN sampled packet. Furthermore, assume 10^6 TCP flows were sampled in total. We compare the above ALL-seq-sflag estimator with the SYN-seq estimator described in Section 4.2. In the above scenario, let $\sigma_i^{(\text{S})} = \sqrt{E}[(\theta_i - \tilde{\theta}_i)^2]$ for the SYN-seq estimator and $\sigma_i^{(\text{A})} = \sqrt{E}[(\theta_i - \tilde{\theta}_i)^2]$ for the ALL-seq-sflag estimator both given by the Cramér-Rao bound. Figure 5 shows the graph of $\sigma_i^{(\text{A})}/\sigma_i^{(\text{S})}$. Note that $1 - \sigma_i^{(\text{A})}/\sigma_i^{(\text{S})}$ is the percentage decrease in the standard deviation error obtained by the All-seq-sflag estimator with respect to the SYN-seq estimator. The graph shows that including all sampled flows with no sampled SYN packet on the estimator improves the estimate of θ_1, $\tilde{\theta}_1$. It also shows little impact on other flow sizes. The above result given through the Fisher information is confirmed by the maximum likelihood estimates from Figures 6 and 7. Also, our experiments show that the ALL-seq-sflag estimator is an *efficient* estimator for the above scenario.

In what follows, we compare the ALL-seq-sflag MLE to the packet count MLE without SYN flag information (using all sampled flows) found in [3] (estimator $f^{(4)}$ according to the nomenclature in [3]) when

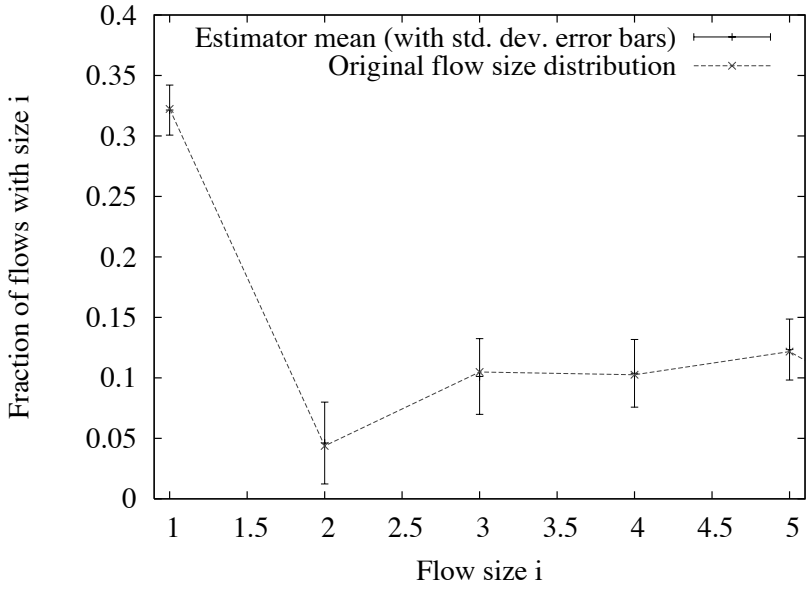

Figure 7: ALL-seq-sflag MLE mean with standard deviation error bars.

applied to TCP flows. We will denote the packet count MLE found in [3] by ALL-pktct MLE.

4.4 Comparison of TCP sequence number MLE to packet counts MLE

The ALL-pktct MLE, as described in [3], uses samples that bear little information about the original flow sizes (as given by the Fisher Information, that for $W > 50$, is close to zero). Therefore the packet counts MLE is unable to provide consistent results when the number of samples is extremely large (by extremely large we mean on the order of billions of samples). The end result is an MLE that is very sensitive to the initial guess of the flow size distribution, which precludes any meaningful comparisons. A good estimator should be fairly robust to initial value of $\tilde{\theta}^{(0)}$.

The following two sets of experiments were performed using synthetic samples drawn from the flow size distribution of the BB-East-1 trace. Let $p = 1/100$ be the sampling rate and $W = 100$ the maximum flow size.

Figure 8 shows the packet count Maximum Likelihood estimates using two distinct initial original flow distribution guesses. An initial guess where flows fractions are uniformly distributed, i.e.,$\tilde{\theta}_i^{(0)} = (1/W)$, $\forall i$, yields curve "ALL-pktct (guess:uniform)" and an initial guess where half of the flows have size one and the remaining flow sizes are uniformly distributed $\tilde{\theta}_i^{(0)} = (1/W)/2$, $\forall i > 1$ and $\tilde{\theta}_1^{(0)} = 0.5$, yields curve "ALL-pktct (guess:notuniform)". Figure 8 depicts both curves with their respective standard deviation error bars. The ALL-pktct MLE is clearly very susceptible to the initial original flow distribution guess. When the initial distribution estimates $\tilde{\tilde{\theta}}^{(0)}$ are not uniform, according to the curve "ALL-pktct MLE (guess:notuniform)", flows of size one account for almost 90% of the original flows with high confidence, an obviously erroneous result. On the other hand, when the initial distribution estimates are uniform as in "ALL-pktct MLE (guess:uniform)", this same metric is 36%, which is fairly close to its true value: 32%. This phenomenon was also observed for the estimator $f^{(4)}$ given in [3].

Under the same scenario, the TCP sequence number and SYN flag MLE yield curves "ALL-seq-sflag (guess:uniform)" and "ALL-seq-sflag (guess:notuniform)", both depicted with their standard deviation error

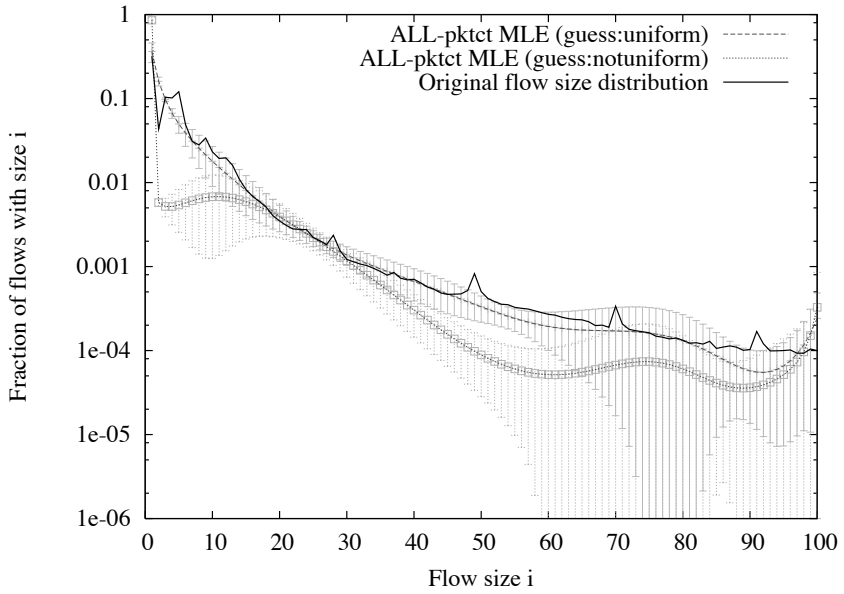

Figure 8: The initial guess $\tilde{\vec{\theta}}^{(0)}$ has a huge impact on quality of the ALL-pktct MLE.

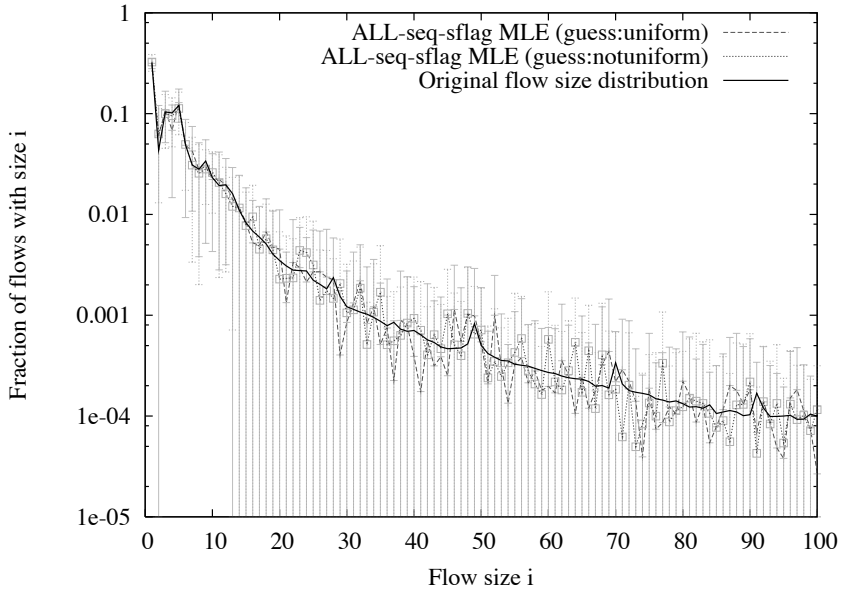

Figure 9: The initial guess $\tilde{\vec{\theta}}^{(0)}$ has a little or no impact on quality of the ALL-seq-sflag MLE.

bars in Figure 9. This result shows that this MLE is clearly more robust to misleading initial distribution estimates $\vec{\theta}^{(0)}$.

The next section is concerned with improving and better evaluating the SYN-seq MLE and ALL-seq-sflag MLE on an actual trace.

5 Evaluation on network traces

The focus of this section is to evaluate the flow size distribution estimators in an Internet backbone environment. We evaluate our algorithms with packet traces from a Tier-1 ISP's backbone network. They are collected from the IPMON, a passive measurement system that captures the first 64 bytes IP packet header of every packet on an optical link [7]. The **BB-East-1** and **BB-East-2** traces are from two OC-48 links between backbone routers on the east coast. The **Access-East** trace is from an access link in the east coast. The statistics of these traces are listed in Table 1.

Internet flows sizes can be on the order of millions of packets, i.e., MLE equation (15) with $W \gg 1$ is intractable. Next we will see how to estimate TCP flow size distributions over real traces for very large maximum flow sizes $W \gg 1$.

Table 1: Trace Facts

Trace	Avg. Rate	Active Flows	Duration
Access-East	373Mbps	61,000/sec	2 hours
BB-East-1	867Mbps	140,000/sec	2 hours
BB-East-2	25Mbps	5,000/sec	2 hours

5.1 Large maximum flow sizes

Unfortunately our model, as presented, requires one parameter for each flow size from 1 to W. One could model the tail of the flow size distribution as a Pareto model, which would replace most of the larger flow sizes parameters by the two parameters of the Pareto distribution. But even in this case, the estimator still needs to compute sample probabilities d_j and this reduces to summing a large number of coefficients (up to W) on equation (2), with its associated computational cost.

Fortunately, TCP sequence number MLEs are fairly robust to mismatches between the modeled maximum flow size W and the actual maximum flow of the set of flows that generated the samples. The estimations presented next were made over real Tier-1 Internet backbone traces. The robustness of the TCP sequence number MLEs on a real sampled trace agrees with our observations made using synthetic traces.

5.2 An approximation to h

Before proceeding to the actual estimation of the flow size distribution we need to address one last issue. Function h introduced in Section 3.3.2 takes as arguments two TCP sequence numbers of two packets in a flow and returns the number of packets sent between these two packets. Before we can estimate flow sizes from real Internet traces we need to approximate h using real Internet sampled flows. We describe this next.

The baseline for our approximation $\tilde{h}(s_1, s_2)$ to $h(s_1, s_2)$ is to use $|s_1 - s_2|$ divided by the maximum data segment transmitted on the flow, where s_1 and s_2 are two TCP sequence numbers of packets belonging

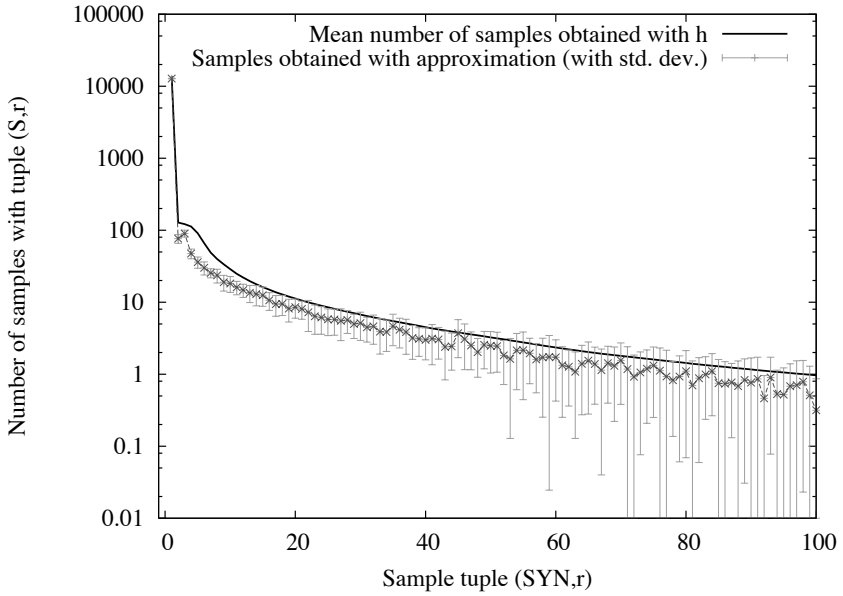

Figure 10: Number of sampled flows labeled with tuples obtained from h drawn synthetically and from $\tilde{h}$ using the real sampled trace. Results obtained from the BB-East-2 trace. Packet sampling rate $p = 0.01$. This graph shows $n\hat{d}_{(\mathrm{S},r)}$, the number of sample tuples (S, r) (from flows with a SYN sampled packet). Notice that the average is slightly underestimated.

to the same flow. The reasoning here is that while a TCP application has enough data to send, most TCP protocol stacks will send packets with data up to the maximum payload size. Most TCP implementations use maximum payload sizes of 1460, 1448 or 536. Notice that we are looking at only one direction of the flow, i.e., we only have access to one side of the two-way TCP connection. Unfortunately a good approximation to h requires some enhancements to the baseline approach.

Zero sized packets and modern web browsers present two difficult issues to resolve finding a good $\tilde{h}$. Zero sized packets will not increase the TCP sequence number counter and, if not sampled, are almost totally invisible to us. Modern web browsers use persistent HTTP 1.1 connections: Suppose an Internet user is reading a news website. A regular user is expected to follow many links on the same web server. Upon requesting a page, the web server will send all packets with the same size *except for the last one*. Then the user's browser will keep the TCP open connection, and in the event of a new user requested page, the browser will ask for more data on this same TCP connection. This creates a TCP flow with more distinct payload sizes than one expects to see in a single TCP connection. One can argue that these are independent TCP flows. The fact is that they share the same SYN packet, which defines a flow in our model.

We first deal with the multiple payload size problem. A sizable amount of the web-servers on the Internet are Linux machines. Linux machines have an interesting behavior on their IPID field, they are all sequential for a given a TCP flow (a reference to the many uses of the IPID field can be found on [1]). With distinct payload sizes inside the same flow, most of them not sampled, $|s_1 - s_2|$ will likely not give us number that is a multiple of the maximum payload size per packet in the flow. If these small sized payloads are not a large fraction of the total number of packets we can verify whether the number of packets obtained using the IPID difference of the packets is close to the number obtained using Sequence Numbers. If so, we will use the IPID difference.

In most TCP flows the majority of the data is sent in one direction, i.e., the TCP sequence number differ-

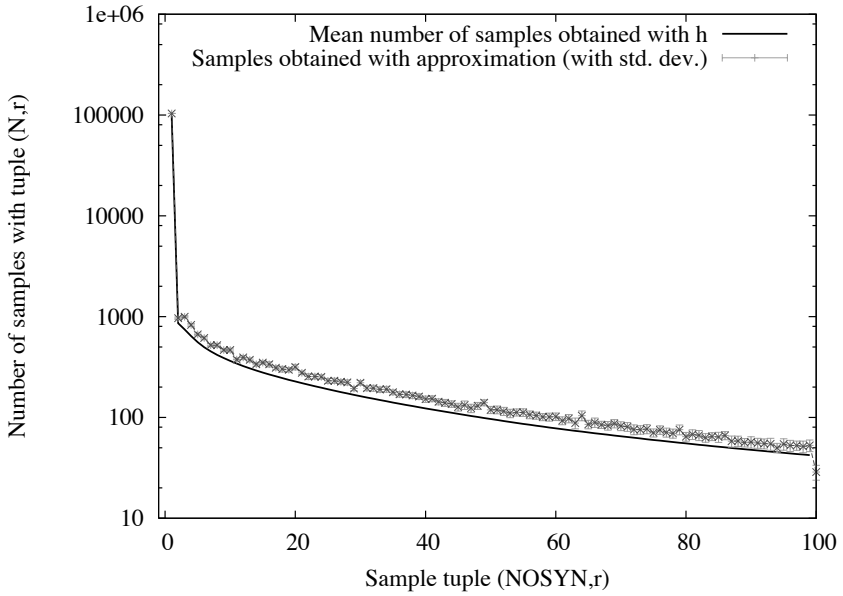

Figure 11: Number of sampled flows labeled with tuples obtained from h drawn synthetically and from $\tilde{h}$ using the real sampled trace. Results obtained from the BB-East-2 trace. Packet sampling rate $p = 0.01$. This graph shows $n\hat{d}_{(\mathrm{N},r)}$, the number of sample tuples (N, r) (from flows without a SYN sampled packet). Notice that the average is slightly overestimated.

ence on one direction is much larger than on the other. If most of the data is being sent in the direction being sampled, the maximum payload size is obtained from the sampled flow, by discarding FIN and SYN packets (usually smaller), and assuming sampled packets are representative of the unsampled packets. Otherwise, we denote the flow as a TCP ACK flow. TCP ACK flows usually have many zero sized packets. One can estimate the value of h on TCP ACK flows by looking at the TCP ACK sequence numbers, which are sequence numbers of the data being sent on the opposite direction of the sampled packets. We keep statistics on the distribution of some specific payload sizes (such as sizes 1460, 536) of non TCP ACK flows and assume that the payload size distribution in both directions is the same. Using the TCP ACK sequence numbers and the above mentioned distribution we obtain an estimate of the value of h.

The above function $\tilde{h}$ is rather simplistic using TCP protocol information; however it seems to work reasonably well although the proposed estimator can certainly benefit from a more accurate model of h. We leave the construction of a good model for h for future research.

The above observations were made from trace Access-East, and then tested on the BB-East-2 trace. Sampling flows on the BB-East-2 trace at rate $p = 1/100$ generates, on average, approximately 125,000 sampled TCP flows to be used by the estimator. Figure 10 (Figure 11) shows how well we can approximate the sample tuples $n\hat{d}_{(\mathrm{S},r)}$ ($n\hat{d}_{(\mathrm{N},r)}$) obtained from $\tilde{h}$ over real sampled data from BB-East-2. Recall that $n\hat{d}_{(\mathrm{S},r)}$ ($n\hat{d}_{(\mathrm{N},r)}$) are the counts of the sampled SYN (NON-SYN) flows where $r = h(s^{(u)}_{max}, s^{(u)}_{min})$.

Note that the use of $\tilde{h}$ results in a slight underestimate of the number of sampled SYN flows and a slight overestimate of the number of sampled NON-SYN flows. This matter needs further investigation but it might be an indicative that sampled flows are suffering from flow splitting [11]. A future research topic is to account for flow splitting in the model.

In what follows we obtain flow size distribution estimates from the BB-East-2 trace.

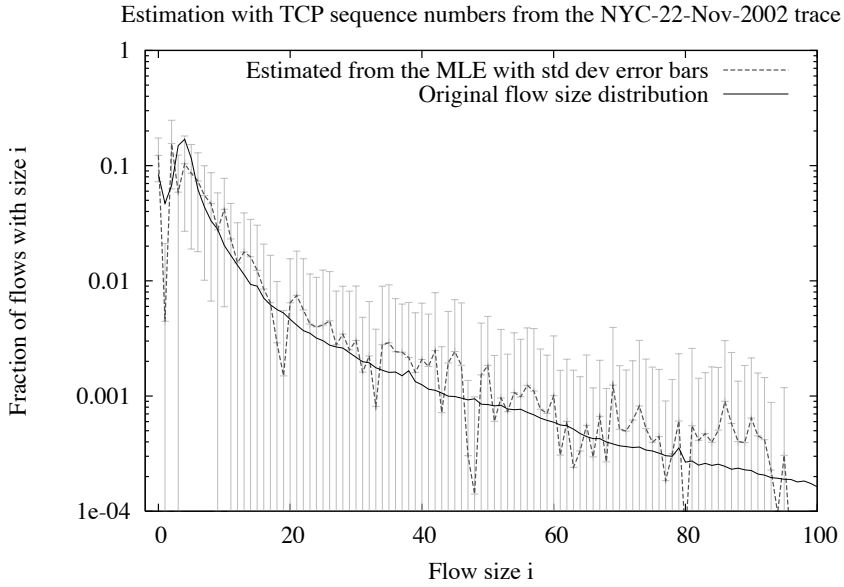

Figure 12: Estimated flow size distribution from the BB-East-2 trace with standard deviation error bars versus the original flow size distribution. Packet sampling rate $p = 0.01$. Using the ALL-seq-sflag MLE.

5.3 Evaluation and performance

Using the sampled flow size distribution obtained using $\tilde{h}$ (Figures 10 and 11), we find estimates for the flow size distribution of the flows contained at the BB-East-2 trace. We use the MLE described in Section 4.3 with maximum flow size $W = 150$. Figure 12 shows that the ALL-seq-sflag MLE can predict flow sizes fairly well. Once again, we use $\tilde{\theta}^{(0)} = 1/W$ as our initial guess estimate.

Although $W = 150$ does not yield a large number of MLE parameters, one would like to be able to run the MLE as fast as possible. At the end of Section 5.1 we saw that it is likely that the ALL-seq-sflag estimator is robust due to the low noise introduced by large flow sizes. In fact, the SYN-seq estimator should be even more robust than the ALL-seq-sflag estimator. The above assumption holds true in practice. Figure 13 show the estimated flow sizes using the SYN-seq MLE over the SYN flow samples of Figure 10. In this model we set the maximum flow size $W = 50$. The Conjugate Gradient algorithm took 85 seconds in average (on a Mobile Pentium4 2.0GHz processor) to achieve the the MLE shown in Figure 13.

In what follows we assess the value of sampling at multiple monitors.

6 Flow Size Estimation on Multiple Monitors

So far we have considered samples from a single monitor. Flows crossing a backbone network will normally cross multiple monitors in the network. In this section we study the value of the information obtained from multiple monitors and how to best use the collected samples at multiple monitors.

The combination of sampled network measurements from multiple monitors was considered in [4]. In [4] the authors focus on estimating $\vec{\theta}$, using local estimates of $\vec{\theta}$, $\vec{\tilde{\theta}}_{(1)}, \ldots, \vec{\tilde{\theta}}_{(m)}$, obtained at m monitors. It assumes all monitors will sample independently. Their goal is to find a new estimate $\vec{\tilde{\theta}}'$ of $\vec{\theta}$ using a linear combination of the local estimates $\vec{\tilde{\theta}}_{(1)}, \ldots, \vec{\tilde{\theta}}_{(m)}$ such that the variance of $\hat{X}'$ is the smallest among all linear combinations. [4] applies this method to obtain reasonable, although not optimal estimation on traffic matrix

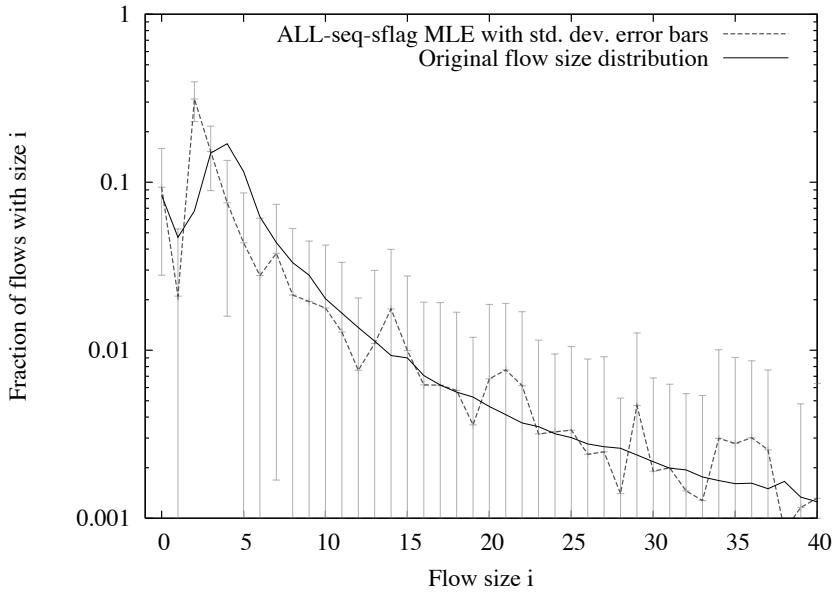

Figure 13: Estimated flow size distribution from the BB-East-2 trace with standard deviation error bars versus the original flow size distribution. Packet sampling rate $p = 0.01$ using the SYN-seq MLE.

information from combined samples. In this section, we will focus on the flow size distribution. Our goal is to determine the information loss from combining local estimates instead of combining all samples and then estimating the desired quantity. This allows us to assess how close the method in [4] is to optimal. In this section we focus on the SYN-seq estimator.

Assume there are u monitors sampling packets at rates $p_1, \ldots, p_u$ respectively and that the same traffic is seen by these u monitors, as in [4]. Let $\mathbf{B}$ be a matrix as defined in Section 2.2 for the TCP sequence number case, with the only change being the sampling rate $p = (1 - \prod_z (1 - p_z))$. This models the case where all packet samples are combined at a single central server and the estimation is performed on the combined samples.

The alternative is to form an estimate at each monitor and then combine them into a single one. This approach was suggested in [4]. Let $W = 200$ be the maximum flow size and $p = 1/64$ be the packet sampling rate when there is only one monitor or $p_1 = p_2 = 1/128$ when there are two monitors. Figure 14 compares the standard deviation of the estimation error of the following three approaches: (1) Estimation using one monitor with sampling rate p, (2) estimation using the combined samples of two monitors at rates p_1 and p_2 and (3) estimation using the combined estimates obtained at each monitor. The results are presented by evaluation both (2) and (3) against (1). Let $\tilde{\tilde{\theta}}^{(\mathrm{L})}$ be the estimates obtained by the approach described in [4]; $\tilde{\tilde{\theta}}^{(\mathrm{SM})}$ be the estimates obtained by the single monitor with sampling rate p; and $\tilde{\tilde{\theta}}^{(\mathrm{TM})}$ be the estimates obtained in a central server from the combined samples collected at the two monitors with sampling rates p_1 and p_2. Let $\sigma_i^{(\mathrm{L})} = \sqrt{E}[(\theta_i - \tilde{\theta}_i^{(\mathrm{L})})^2]$, $\sigma_i^{(\mathrm{SM})} = \sqrt{E}[(\theta_i - \tilde{\theta}_i^{(\mathrm{SM})})^2]$ and $\sigma_i^{(\mathrm{TM})} = \sqrt{E}[(\theta_i - \tilde{\theta}_i^{(\mathrm{TM})})^2]$ obtained by the Cramér-Rao bound. Figure 14 shows the graph of curve $\sigma_i^{(\mathrm{L})}/\sigma_i^{(\mathrm{SM})}$ ("Combining independent estimates") against curve $\sigma_i^{(\mathrm{TM})}/\sigma_i^{(\mathrm{SM})}$ ("Combining samples"). The results show that combining the sample at a central server is almost as good as sampling once with double the rate. On the other hand, combining the two independent estimates increases the standard deviation error of the estimates in 50%.

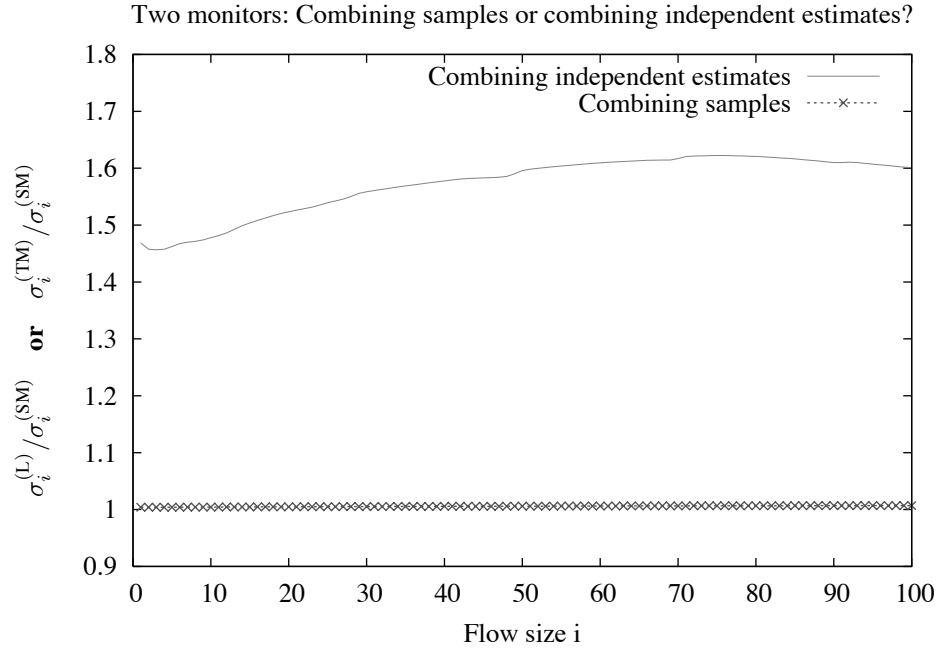

Figure 14: This graph uses the Cramér-Rao bound to show the advantage of making the estimation with the combine samples taken at each monitor over the combination of the two independent estimations taken at each monitor. The evaluation is done over distribution from trace BB-East-2.

7 Conclusions and Future Work

In this paper we have focused on a key issue that arises when conducting measurements for the purpose of estimating network statistics such as the flow size distribution, namely, what are the values of different types of information on the quality of the estimate? Using flow size distribution as an example and packet sampling as the measurement technique, we studied the values of different informations, packet counts, SYN information, and sequence number information. Using the Fisher information through its application via the Crámer-Rao bound on mean squared error, we found that both SYN information and sequence number information each can generate a substantial reduction in the estimation error. Using this as a starting point, we presented MLEs based on the conjugate gradients method, which achieve close to the Crámer-Rao bound, even for small sample sizes. We also explored the benefit of including packet counts for both flows with and without SYN information, and determined that the former is useful in reducing the errors associated with estimating the probability of flows of size one. Last, we applied the framework to determine the benefits of combining observations from multiple monitoring sites. Our analysis shows substantial benefit in performing estimation on the combined set of observations as opposed to combining the estimates from made on observations at individual monitoring sites.

This is a first step in an attempt to understand the value of different types of information for the purpose of estimating network statistics. Our future work will focus both on applying our framework to other estimation problems and more specifically to refining the application to flow size distribution estimation. For example, there is a need for a parsimonious model of the flow size distribution with a small number of parameters. Another research direction is to extend the work on multiple monitors. For example, can one use the Fisher information to derive an adaptive mechanism for determining sampling rates at different monitors so as to minimize error subject to a resource constraint.

8 Acknowledgments

We acknowledge the fundamental contribution of Darryl Veitch on suggesting the TCP sequence number field as a high informational protocol field for the flow size estimation.

This work has been supported in part by NSF under grant ANI-0325868, the CAPES Brazilian agency award 2165031 and the Sprint Advanced Technology Laboratories. The equipment was supported in part by NFS RI infrastructure grant under award number EIA-0080119. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] Weifeng Chen, Yong Huang, Bruno F. Ribeiro, Kyoungwon Suh, Honggang Zhang, Edmundo de Souza e Silva, Jim Kurose, and Don Towsley. Exploiting the IPID field to infer network path and end-system characteristics. In *Proceeding of the 2005 Passive and Active Measurement (PAM'05) Workshop*, March 2005.

[2] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & sons, 1991.

[3] Nick Duffield, Carsten Lund, and Mikkel Thorup. Estimating flow distributions from sampled flow statistics. *IEEE/ACM Transactions on Networking*, 13(5):933–946, 2005.

[4] Nick Duffield, Carsten Lund, and Mikkel Thorup. Optimal combination of sampled network measurements. In *IMC 05́: Proceeding of the 5th ACM/USENIX Internet Measurement Conference*, October 2005.

[5] Nick G. Duffield, Carsten Lund, and Mikkel Thorup. Learn more, sample less: control of volume and variance in network measurement. 51(5):1756–1775, 2005.

[6] Cristian Estan, Stefan Savage, and George Varghese. Automatically Inferring Patterns of Resource Consumption in Network Traffic. In *Proc. ACM SIGCOMM '03*, Karlsruhe, Germany, Aug. 2003.

[7] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and C. Diot. Packet-level traffic measurements from the sprint IP backbone. *IEEE Network*, 2003.

[8] John D. Gorman and Alfred O. Hero. Lower bounds for parametric estimation with constraints. *IEEE Transactions on Information Theory*, 36(6):1285–1301, Nov 1990.

[9] Nicolas Hohn and Darryl Veitch. Inverting sampled traffic. In *IMC '03: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, pages 222–233, New York, NY, USA, 2003. ACM Press.

[10] Steven M. Kay. *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*. Prentice Hall PTR, March 1993.

[11] Ramana Rao Kompella and Cristian Estan. The power of slicing in internet flow measuremet. In *IMC '05: Proceeding of the 5th ACM/USENIX Internet Measurement Conference*, October 2005.

[12] S. Muthukrishnan. Data streams: algorithms and applications. In *Proc. of ACM SODA, invited talks*, pages 413–413, 2003. A complete version is available at `http://athos.rutgers.edu/~muthu/stream-1-1.ps`.

[13] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C : The Art of Scientific Computing*. Cambridge University Press, October 1992.

[14] Richard A. Redner and Homer F. Walker. Mixture Densities, Maximum Likelihood and the EM Algorithm. *SIAM Review*, 26(2):195–239, April 1984.

[15] RFC791. Internet protocol. September 1981. DARPA Internet Program Protocol Specification.

[16] M. J. Schervish. *Theory of Statistics*. Springer, 1995.

[17] Ram Zamir. A Proof of the Fisher Information Inequality via a Data Processing Argument. *IEEE Transactions on Information Theory*, 44(3):1246–1250, 1998.

[18] Qi Zhao, Abhishek Kumar, Jia Wang, and Jun Xu. Data streaming algorithms for accurate and efficient measurement of traffic and flow matrices. In *Proc. of ACM SIGMETRICS*, June 2005. to appear.

[19] Cisco NetFlow. http://www.cisco.com/warp/public/732/Tech/nmp/netflow.

[20] IPFIX, IETF Working Group Charter IP Flow Information Export. http://www.ietf.org/html.charters/ipfix-charter.html.

[21] Packet Sampling, IETF Working Group Charter PSAMP. http://www.ietf.org/html.charters/psampcharter.html.

[22] sFlow. http://www.sflow.org.

[23] Sprint packet trace analysis. http://ipmon.sprint.com/packstat.