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

Bruno Ribeiro, Don Towsley
Department of Computer Science
University of Massachusetts at Amherst
140 Governors Drive
Amherst, MA 01003-9264
{ribeiro,towsley}@cs.umass.edu

Tao Ye, Jean Bolot
Sprint ATL
One Adrian Court
Burlingame, CA 94010
{Tao.Ye,Bolot}@sprint.com

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. In this paper we take a systematic approach, using the Fisher information metric and the Cramér-Rao bound, to understand the contributions that different types of information within sampled packets have on the quality of flow-level estimates. We provide concrete evidence that, without protocol information and with packet sampling rate $p = 0.005$, any accurate unbiased estimator needs approximately 10^{16} sampled flows. The required number of sampled flows drops to roughly 10^4 with the use of TCP sequence numbers. Furthermore, additional SYN flag information significantly reduces the estimation error of 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 using 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.

Categories and Subject Descriptors

H.1.1 [**Systems and Information Theory**]: Value of information; C.2.3 [**Network Operations**]: Network monitoring; G.3 [**Probability and Statistics**]: Nonparametric statistics

General Terms

Measurement, Theory

Keywords

Probabilistic Sampling, Packet Sampling, Flow Size Distribution, Fisher Information, Efficient Estimator, Maximum Likelihood Estimation

*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.

IMC'06, October 25–27, 2006, Rio de Janeiro, Brazil.

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 the 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 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 should apply to other metrics as well, such as traffic matrix estimation. 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 being 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 are high in information content value beneficial to flow size estimation. We show TCP protocol information to be essential for accurate unbiased flow size estimators. Further we bring the first study to our knowledge of the benefits of computing flow size distribution estimates by combining samples from multiple monitors. We 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 are 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].

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 from multiple monitors in Section 6. Finally we conclude with Section 7.

2. MODELING 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. 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 packet 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. Table 1 summarizes most of the definitions used throughout this paper.

Notation	Definition
ALL-pktct	Estimator that uses packet counts from all sampled flows.
SYN-pktct	Estimator that uses packet counts from SYN sampled flows.
SYN-seq	Estimator that uses TCP sequence numbers from SYN sampled flows.
ALL-seq-sflag	Estimator that uses TCP sequence numbers and SYN flags from all sampled flows.
$W \geq 2$	Maximum flow size
$0 < p < 1$	Packet sampling rate (in samples per packet).
$i \in \{1, \ldots, W\}$	Flow size before sampling (*original* flow size).
$\vec{\theta} = [\theta_i]$	True flow size distribution.
$\vec{\theta}' = [\theta'_i]$	Flow size distribution $\vec{\theta}$ conditioned on at least one of its packets being sampled.
$\tilde{\vec{\theta}} = [\tilde{\theta}_i]$	Estimated flow size distribution.
$j \in \mathcal{L}$	Sample label.
$\vec{d} = [d_j]$	Distribution of the sampled flows.
$\mathbf{B} = [b_{i,j}]$	$b_{i,j}$ is the probability that a sampled flow, with original flow size i, has sample label j.
n	Number of sampled flows.
$\hat{d}_j^{(n)}$	Fraction of the n sampled flows with label j.
$\alpha^{(n)}$	Likelihood function for n sampled flows.
$h(a, b)$	Given two TCP sequence numbers a and b from two packets of the same flow, $h(a, b)$ returns the number of packets from the same flow sent between these two packets.

Table 1: Notations table.

2.1 Basic Model

Assume the original flow size is upper bounded by $W \geq 2$.

Let θ_i be the fraction of original flows of size i that cross the monitor during some given time interval and let θ'_i be the fraction of original flows of size i that were sampled. 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 conditional distribution of $\vec{\theta}$ conditioned at least on one of its packets being sampled [3]. Under the Bernoulli sampling process assumption, $\vec{\theta}$ and $\vec{\theta'}$ are related as follows:

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

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

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

where $b_{i,j}$ 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

$$\vec{d} = \mathbf{B}\vec{\theta}, \quad (3)$$

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 n be the number of sampled flows and $\hat{D}_j^{(n)}$, $j \geq 0$, denote the total number of sampled flows with j sampled packets. We can also further define

$$\hat{\vec{d}}^{(n)} = [\hat{d}_j^{(n)}] = [\hat{D}_j^{(n)}/n]. \quad (4)$$

An estimator without protocol information. In [3] the authors present a set of estimators based on the above samples, i.e., without TCP protocol information. We refer to an estimator without protocol information as an "ALL-pktct" estimator. "ALL" refers to the use of all TCP sampled flows. And "pktct" refers to an estimator that uses only packet counts.

Next we extend the model to account for protocol information, particularly TCP SYN flags and sequence numbers.

2.2 TCP SYN flag 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.

SYN sampled flows. 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 SYN sampled flow. Assume there are n sampled flows. We use $\hat{d}_{(\mathrm{S},m)}^{(n)}$ to denote the fraction of the n sampled flows where a SYN packet is sampled and there are m sampled packets. Likewise, we denote by $\hat{d}_{(\mathrm{N},m)}^{(n)}$ the fraction of the n sampled flows where there was no SYN sampled packet and there are m 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 show how to add flows sampled without SYNs to the estimator. Equation (3) holds for SYN sampled flows with matrix $\mathbf{B}$ properly redefined. The modification to $\mathbf{B}$ is found in [3]. We refer an estimator that uses $\hat{d}_{(\mathrm{S},m)}^{(n)}$ 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 payload bytes in a flow. Assume that the TCP start sequence number, i.e., 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 using 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 of the same flow and returns the number of packets sent between a and b including a and b. We acknowledge that it is not easy to construct a function h that returns the exact packet counts. In Section 5 we 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 $\hat{d}_r^{(n)}$ 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 $b_{i,(\mathrm{S},r)} = p(1-p)^{i-r}$, $\forall i \leq W, 2 \leq r \leq i$ and $b_{i,(\mathrm{S},1)} = (1-p)^{i-1}$, and the rest of the matrix being zero. Finally, let us denote an estimator that uses $\hat{d}_{(\mathrm{S},r)}^{(n)}$ as a "SYN-seq" estimator.

Samples with TCP sequence numbers and full SYN flag information. Our traces show that only 20% of the TCP sampled flows contain a SYN sampled packet. In [3] the authors conjecture that there are few SYN sampled flows which implies less accurate estimates. Next we increase the number samples by adding sampled flows without a SYN sampled packet to the estimator.

Let us denote the estimator that uses TCP sequence numbers and SYN flags (SYNs and non SYN sampled flows) as a "ALL-seq-sflag" estimator. Let $\mathbf{B}$ denote the sampling probability matrix as defined by (3). Let j denote a tuple $(\mathrm{SYNFLAG}, r)$, where $r = h(s_{max}^{(u)}, s_{min}^{(u)})$. Let SYNFLAG = S when there is a SYN packet in the sampled flow and SYNFLAG = N otherwise. Thus $b'_{i,<\mathrm{S},r>} = p\,(1-p)^{i-r}$ and $b'_{i,<\mathrm{N},r>} = (i-r)\,p\,(1-p)^{i-r}$. The element i,j of matrix $\mathbf{B}$ is $b_{i,j} = b'_{i,j}/\sum_{\forall j} b'_{i,j}$.

We have introduced several types of information. One question that remains is which type of information is valuable to an 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 $E[T(\theta_i)] = \theta_i$.

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 a way to compute a tight lower bound on the mean squared error for flow size estimators.

3.1 Measuring information: Fisher information

The Fisher information can be thought of as the amount of information that a set of observable samples, $\hat{\vec{d}}^{(n)}$, carry about unobservable parameters $\vec{\theta}$ or $\vec{\theta}'$ upon which the probability distribution of the samples depends. The results in this section derived for $\vec{\theta}$ are also valid for $\vec{\theta}'$.

The Fisher information is 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 information of each of the samples [2]. The above result applies to sampled flows as follows:

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

PROOF. If packets are sampled independently 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]. □

Next we compute the Fisher information of a single sampled flow. Assume maximum flow size $W \geq 2$ and $\theta_i > 0$ with $1 \leq i \leq W$. Let $n\hat{d}_j^{(n)}$ 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 $\hat{d}_j^{(1)} = 0$ for all $j \neq j'$ and $\hat{d}_{j'}^{(1)} = 1$. Define an operator $(\cdot)_j$ over a vector that retrieves the element indexed by sample label j. Let

$$\alpha(\hat{\vec{d}}^{(1)}; \vec{\theta}) = \sum_{\forall j \in \mathcal{L}} \hat{d}_j^{(1)} (\mathbf{B}\vec{\theta})_j = \sum_{\forall j \in \mathcal{L}} \hat{d}_j^{(1)} d_j \qquad (5)$$

be the conditional probability that this sampled flow has sample label j' for a given flow size distribution $\vec{\theta}$. Function 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 α are constrained by:

$$\sum_{\forall i} \theta_i = 1 \qquad (6)$$

and

$$0 < \theta_i < 1, \ \forall i. \qquad (7)$$

Unfortunately the Fisher information as defined in [2] is unconstrained. But constraints (7) can be included by a simple change of variables in α:

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

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 $\beta(\vec{\gamma})$, a vector whose i-th element is $\beta(\gamma_i)$, then the likelihood function f of one sampled flow is

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

Under the above conditions we find the Fisher information 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 also include constraint (6). Note that d_j is equal to $P(\hat{d}_j^{(1)} = 1)$, the probability that our sole sampled flow has sample label j. Let

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

also with

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

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 of $f(\vec{d}^{(1)}; \vec{\gamma})$. We obtain $\mathcal{I}$ from its inverse $\mathcal{I}^{-1}$. The inverse of the Fisher information with $\vec{\gamma} \in \mathcal{D}$, $\mathcal{I}^{-1}(\vec{\gamma})$, is a $W \times W$ matrix

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

where $\mathbf{G}^{\mathrm{T}}$ is the transpose of $\mathbf{G}$.

The Fisher information can be used to compute a lower bound on the mean squared 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 of the Fisher information [8], provided some regularity conditions required by the Cramér-Rao bound. These regularity condition [10] translates into $\sum_j \partial d_j / \partial \gamma_i = \partial / \partial \gamma_i \sum_j d_j$, $\forall i$ 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

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

with $\mathcal{I}^{-1}$ as defined in (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.

Definitions	Number of sampled flows needed
ALL-pktct	$> 2.25 \times 10^{16}$
SYN-pktct	$> 3.4 \times 10^{16}$
ALL-seq-sflag	4×10^4

Table 2: Number of sampled flows an unbiased estimator needs in order to achieve standard deviation error of 0.1 for flows of size one. Results for maximum flow size $W = 20$ and sampling rate $p = 1/200$ over the BB-East-1 trace flow size distribution.

Although 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 $\vec{\tilde{\theta}}$ is

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

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

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

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 Section 3.3.2 displays the use of the Fisher Information through the Cramér-Rao bound, in designing better estimators. In the next two examples we will look at SYN sampled flows. The parameters of the following two examples are just to illustrate the use of the Fisher information. On Section 4 we will look at more realistic scenarios.

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 $(\mathbf{G})_i = \theta_i^2/(\theta_i - 1)$. Let β^{-1} be the inverse of function β. 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 $\vec{e}_j = (b_{1,j}, b_{2,j}) \cdot (\vec{\theta}^2/(\vec{\theta} - 1))$. Let j denote the number of sampled packets in a SYN sampled flow. Then $b_{1,1} = 1$, $b_{1,2} = 0$, $b_{2,1} = 0.99$ and $b_{2,2} = 0.01$. The 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 as seen in Section 3.2. Matrix $\mathbf{H}$ is

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

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 Essential 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 data processing theorem [17] states that adding information can only increase the Fisher information. Thus, we expect that an efficient estimator using extra information performs better, or at least no worse, than an efficient estimator that does not use the extra information. This clearly holds as one can always throw the extra information away inside the estimator.

For our next example, assume a maximum flow size $W = 4$ and $\vec{\theta} = (0.56, 0.08, 0.18, 0.18)$. The elements of $\mathbf{B}$ are $b_{i,1} = (1-p)^{i-1}$ and $b_{i,j} = p(1-p)^{i-j}$ for $j > 1$. We compute the Cramér-Rao bound for a sampling rate of $p = 1/100$. Also consider the estimation using packet counts over SYN flows (SYN-pktct) as defined in Section 2.2. Figure 1 shows the Cramér-Rao bound obtained with 10^8 sampled flows 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. The graph also shows that the SYN-pktct estimator is able to gather very little information about the original flow sizes. Next we look at an example where the Fisher information is used to obtain the number of samples needed by a given estimate error using different types of protocol information.

3.4 Minimum number of samples required for high quality estimates

The Fisher information is also a powerful tool to adjust measurement parameters. Through the Cramér-Rao bound, one can assess the minimum number of samples needed to achieve a given error. In the following example we use $W = 20$ and $p = 1/200$ and calculate how many samples are needed until the best unbiased estimator can achieve a mean standard deviation error of 0.1 for flows of size one. In the following experiment we renormalized the flow size distribution obtained from the Sprint backbone network. The flow size distribution renormalization creates a distribution $\vec{\theta}$ that is a re-scaled true Internet flow size distribution but with maximum flow size W. The original distribution comes from the trace BB-East-1, summarized in Table 3 at the beginning of Section 5. The results show that without any protocol information, only using packet counts (ALL-pktct), the best unbiased estimator needs at least 2.25×10^{16} sampled flows. Using SYN sampled flows and packet counts (SYN-pktct) the best unbiased estimator needs at least 3.4×10^{16} sampled flows (from where 7.5×10^{15} are SYN sampled flows). On the other hand, the best unbiased estimator using SYN flags and TCP sequence numbers, ALL-seq-sflag, needs a dramatically lower number: 4×10^4 sampled flows. These findings are summarized in Table 2.

Next we shortly present MLEs for the ALL-pktct, SYN-pktct, SYN-seq and ALL-seq-sflag estimators. Experimentally we will find that the ALL-seq-sflag MLE is efficient in that it approaches the Cramér-Rao bound even for a small sample size, $n = 260,000$.

4. FINDING AN OPTIMAL UNBIASED ESTIMATOR

The Maximum log-Likelihood Estimator (MLE), finds a

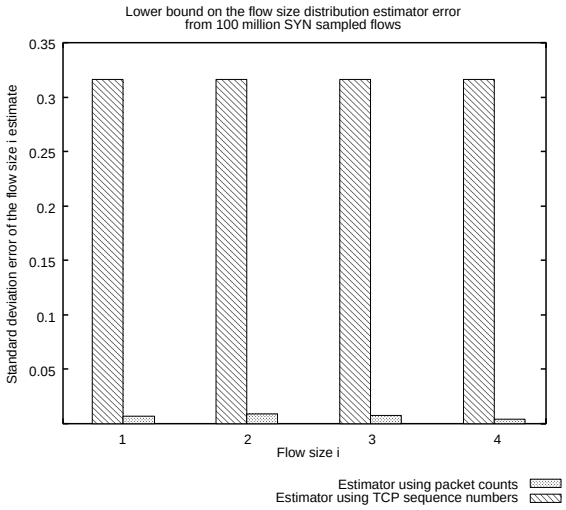

Figure 1: Cramér-Rao bounds of the examples on Section 3.3.2. This graph compares the estimation of SYN-pktct to the SYN-seq. Notice that adding TCP sequence numbers to the estimation greatly improves its quality.

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 MLEs for the models in Section 2. In particular we show that the ALL-seq-sflag MLE does not require a large number of samples to be unbiased and 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 the first part of this section we estimate the flow size distribution using only SYN sampled flows. This, of course, does not account for the "noise" introduced by flow-splitting, which splits one long original flow into two or more shorter ones. We will not account for flow splitting, although [11] shows that is possible to do so. We will evaluate the complete model with "noise" in Section 5 on an actual trace. Next we introduce the MLE for our model.

4.1 MLE with conjugate gradients

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

$$\tilde{\vec{\theta}} = \arg\max_{\tilde{\vec{\theta}}} \; n \sum_{\forall j \in \mathcal{L}} \hat{d}_j^{(n)} \ln(\mathbf{B}\tilde{\vec{\theta}})_j \qquad (15)$$

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 $\hat{D}_{(\mathrm{S},r)}^{(n)}$ denote the number of SYN sampled flows with label r sampled packets. Let $\hat{d}_{(\mathrm{S},r)}^{(n)}$ 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 $\hat{\tilde{\theta}}^{(n)}$ by the successive refinement of previous estimates:

$$\tilde{\theta}_i^{(k+1)} = \tilde{\theta}_i^{(k)} \sum_{\forall j \in \mathcal{L}} \frac{b_{i,j} \hat{d}_{(\mathrm{S},j)}^{(n)}}{\sum_{r=1}^{W} \tilde{\theta}_r^{(k)} b_{r,j}},$$

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 felt more by EM [14] (conjecture strengthened by our practical experience with our EM and conjugate gradients method implementations when applied to the flow size estimation problem).

We instead use the method of conjugate gradients [13] to compute a solution to (15). Our conjugate gradients MLE algorithm was implemented with the help of the wnlib library[1].

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

$$\frac{\partial}{\partial \theta_i} \ln \alpha^{(n)}(\hat{\vec{d}}^{(n)}; \tilde{\vec{\theta}}) = \sum_{\forall j \in \mathcal{L}} \frac{b_{i,j}\, \hat{d}_j^{(n)}}{\sum_{r=1}^{W} \tilde{\theta}_r\, b_{r,j}} - 1.$$

The remaining constraints $0 < \tilde{\theta}_i < 1$, $\forall i \in \{1, \ldots, W\}$ 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 can have zero probability.

4.2 The use of TCP sequence numbers on SYN sampled flows

The following two experiments, with results shown in Figures 2 and 3, were designed to compare the use of various types of information on the MLE accuracy. Let $W = 50$ be

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

the maximum flow size and $p = 1/200$ be the packet sampling rate. We use samples from a renormalized flow size distribution obtained from the Sprint backbone network. The renormalized flow size distribution is based on the distribution of the BB-East-1 trace, summarized at the beginning of Section 5 in Table 3. These experiments use 10^{12} synthetically generated flows that, in average, produce 1.8×10^{10} sampled flows after packet sampling (from where 5×10^9 are SYN sampled flows). The initial value for the MLE optimization is $\tilde{\theta}_i^{(0)} = 1/W$.

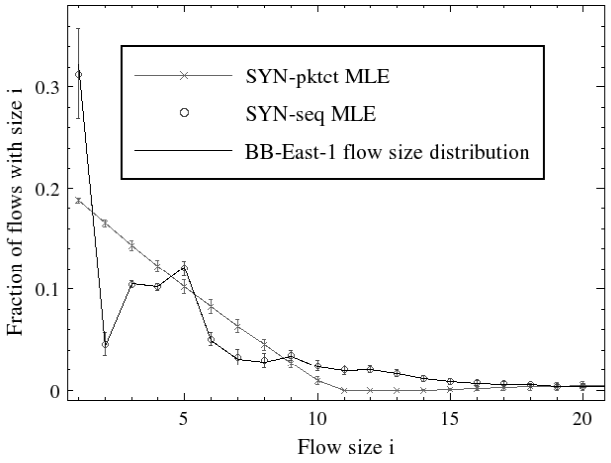

Figure 2: Flow size distribution estimate obtained with SYN-pktct and SYN-seq MLEs. Obtained using 120 runs with 5×10^9 SYN sampled flows each and $p = 1/200$ as the sampling rate. Clearly the SYN-pktct MLE is not an unbiased estimator and cannot capture important features of the original flow size distribution. On the other hand, the SYN-seq MLE is not only an unbiased estimator but also has small standard deviation errors for most flow sizes.

Figure 2 shows a graph with the original flow size distribution and its estimates using SYN-pktct and SYN-seq MLEs. The MLE curves show the mean and standard deviation errors of the estimates obtained from 120 independent runs. The standard deviation errors are computed with respect to the average of the estimates and not to the true flow size distribution values $\vec{\theta}$. Observe that although SYN-pktct MLE exhibits tight standard deviation errors, most estimates are wrong. Moreover, it does not capture important features of the original flow size distribution. Wrong estimate averages and small standard deviation errors in the SYN-pktct MLE are indications that the estimator was able to extract only a small amount of information from the 5×10^9 SYN sampled flows. On the other hand, the SYN-seq MLE, which adds TCP sequence number information to the previous estimator, shows itself to be an unbiased estimator (i.e., $E[\tilde{\theta}_i] = \theta_i\,,\ 1 \le i \le W$) with tight standard deviation errors. According to the Cramér-Rao bound, the SYN-pktct MLE needs at least 10^{17} SYN samples flows to achieve a result comparable to the one obtained using the SYN-seq MLE with 5×10^9 SYN sampled flows.

Note that both SYN-pktct and SYN-seq estimators neglected all sampled flows without a SYN sampled packet, which amounts to 80% of the sampled flows in the BB-East-1 trace. The estimator accuracy could be increased by adding the remaining 80% of the sampled flows to the estimator. In [3] the authors argue that there are not enough SYN flows to find good estimates using the SYN-pktct MLE. In what follows we consider all sampled flows and show that the best estimator in [3], "ALL-pktct MLE" according to our nomenclature, also suffers from the same problems as the SYN-pktct MLE. We further show that adding flows without a SYN sampled packet drastically increases the accuracy of the estimator that uses TCP sequence numbers.

4.3 MLEs using all sampled flows

Incorporating SYN flag information for all sampled flows 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 ALL-pktct MLE presented in [3]. However we will restrict this modification to the estimator with TCP sequence numbers further referred as "ALL-seq-sflag estimator". In this section we compare the ALL-seq-sflag MLE to the ALL-pktct MLE.

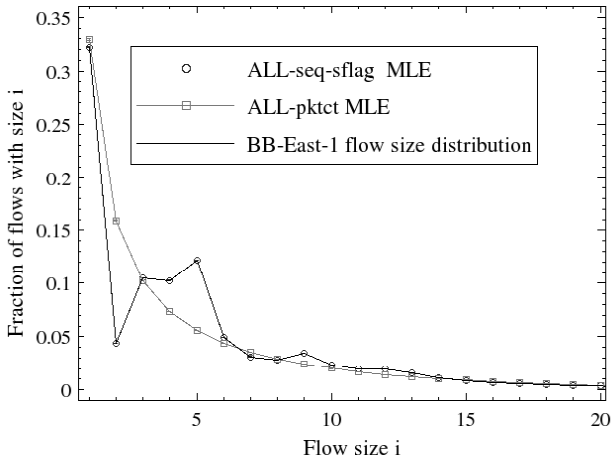

Figure 3: Flow size distribution estimates obtained from ALL-pktct and ALL-seq-sflag MLEs. We used 120 independent runs with 1.8×10^{10} sampled flows each and $p = 1/200$ as the sampling rate. As expected, 1.8×10^{10} sampled flows is too small of a sample set for the ALL-pktct MLE to become an unbiased estimator. The SYN-seq-sflags MLE, on the other hand, substantially improved on the SYN-seq MLE estimates.

The following experiment uses the parameters given at the beginning of Section 4.2. Figure 3 shows graphs of the original flow size distribution and its estimates obtained from ALL-pktct and ALL-seq-sflag MLEs. Again the estimator using packet counts (ALL-pktct) was not unbiased and unable to capture important features of the flow size distribution. It is interesting to note that the ALL-pktct MLE seems to find an accurate estimate for the fraction of flows with size one. This accuracy is misleading. On our experiments we found that this "accuracy" is highly dependent on the initial value $\tilde{\vec{\theta}}^{(0)}$ of the MLE algorithm. We also found that ALL-pktct MLE estimates $E[\tilde{\theta}_1] = 0.32$ with

high confidence (the standard deviation errors on the graph are very small) for the BB-East-2 trace, result similar to the one obtained on Figure 3 for the BB-East-1 trace. But in the BB-East-2 case, the true value is $\theta_1 = 0.09$, while in the BB-East-1 trace $\theta_1 = 0.36$. This means that estimates from ALL-pktct and SYN-pktct do not have a strong relationship with the true distribution values.

Figure 3 also shows that the ALL-seq-sflag MLE estimates are unbiased and have very tight standard deviation errors for the BB-East-1 trace. Similar results were also obtained from the BB-East-2 trace. Note that ALL-seq-sflag MLE greatly improves on the SYN-seq MLE estimates for small flow sizes. Next we show that ALL-seq-sflag MLE can be considered as an efficient estimator and that the Cramér-Rao bound is tight.

4.4 An efficient estimator: ALL-seq-sflag MLE

Figure 4 shows the mean standard deviation error of ALL-seq-sflag MLE estimates compared to its respective Cramér-Rao bound. For a large number of sampled flows ($\geq 10^8$) the Cramér-Rao bound and the ALL-seq-sflag MLE mean standard deviation error are almost indistinguishable. Thus the Cramér-Rao bound is tight and the ALL-seq-sflag MLE appears to be *efficient* for a large number of samples. For a much smaller sample set, 260,000 sampled flows, there is a small bias on the estimates of the ALL-seq-sflag MLE. The mean standard deviation error that is fairly close to the Cramér-Rao bound. Thus, one can argue that the ALL-seq-sflag MLE is an *efficient* estimator for practical purposes even when there are only 260,000 sampled flows.

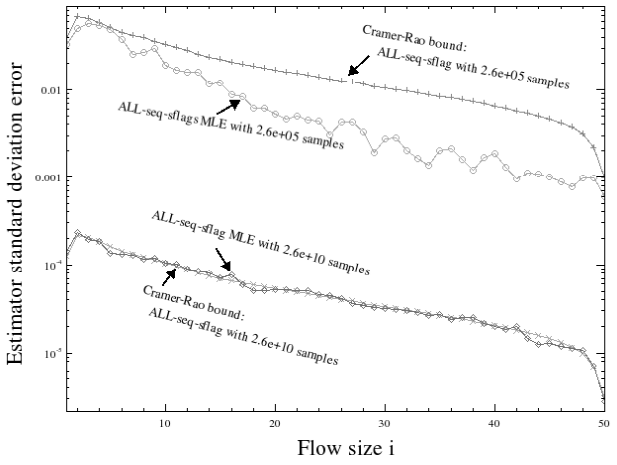

Figure 4: This graph compares the standard deviation errors of the ALL-seq-sflag MLE estimates with the Cramér-Rao bound.

The next section evaluates the SYN-seq 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 using IPMON, a passive measurement system that captures the first

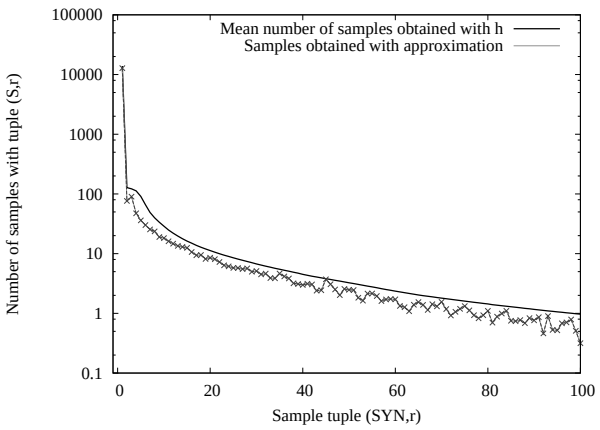

Figure 5: Number of sampled flows with label $(\mathbf{S}, r), r \geq 1$ obtained from both h drawn synthetically, and $\tilde{h}$ obtained using the real sampled trace. Results from the BB-East-2 trace. Packet sampling rate $p = 0.01$. This graph shows $n\hat{d}_{(\mathbf{S},r)}$, the number of sample tuples $(\mathbf{S}, r)$ (from flows with a SYN sampled packet). Notice that the average is slightly underestimated.

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 3.

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 3: 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 in Section 2, 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 implies 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. All estimates presented next were from the BB-East-2 trace.

5.2 An approximation to $h(s_{min}, s_{max})$

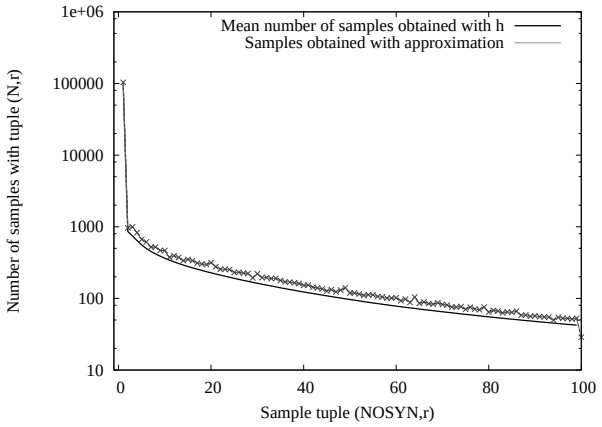

Figure 6: Number of sampled flows with label $(\mathbf{S}, r), r \geq 1$ obtained from both h drawn synthetically, and $\tilde{h}$ obtained using the real sampled trace. Results from the BB-East-2 trace. Packet sampling rate $p = 0.01$. This graph shows $n\hat{d}_{(\mathbf{N},r)}$, the number of sample tuples $(\mathbf{N}, r)$ (from flows without a SYN sampled packet). Notice that the average is slightly overestimated.

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 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 of h requires enhancements to the baseline approach.

Zero sized packets and modern web browsers present two difficult issues to resolve in finding a good $\tilde{h}$. (1) Since zero sized packets do not increase the TCP sequence number counter, they are almost totally invisible to us if not sampled. (2) Modern web browsers use persistent HTTP 1.1 connections since an user is expected to follow many links on the same web server. Upon receiving a request for a page, the web server sends all packets with the same size *except for the last one*. The user's browser keeps the TCP connection open, and in the event of a new user requested page, it asks for more data over the same TCP connection. This creates a TCP flow from possibly many independent flows. One can argue that these are independent TCP flows and should be treated as such. However, as they share the same SYN packet, our model groups them into a single flow.

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 a 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 difference on one direction is much larger than on the other. If most of the data is being sent in the direction being sampled, we obtain maximum payload sizes from the sampled flow, by discarding FIN and SYN packets (usually smaller), 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 a rather simplistic application of TCP protocol information; however it works reasonably well although the proposed estimator can certainly benefit from a more accurate model of h. We leave the construction of a better model for future research.

The above observations were made from trace Access-East, and then tested on BB-East-2. 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. Figures 5 and 6 show how well we can approximate the sample tuples $n\hat{d}_{(\mathrm{S},r)}$ and $n\hat{d}_{(\mathrm{N},r)}$, respectively, 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 indicate 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.

5.3 Evaluation and performance

Using the sampled flow size distribution obtained using $\tilde{h}$ (Figures 5 and 6), we find estimates for the flow size distribution of the flows contained at the BB-East-2 trace. We use the SYN-seq MLE with maximum flow size $W = 50$. Figure 7 shows that the SYN-seq MLE captures some features of the flow size distribution. Once again, we use $\tilde{\theta}^{(0)} = 1/W$ as the initial guess estimate for the MLE algorithm. The conjugate gradient algorithm took 85 seconds in average (on

a Mobile Pentium4 2.0GHz processor) to achieve the the estimates shown in Figure 7. The ALL-seq-sflag MLE with $W = 150$ obtains similar but noisier results. The reason why SYN-seq MLE outperforms the ALL-seq-sflag MLE is the subject of further investigation. Flow splitting could be one of the possible causes.

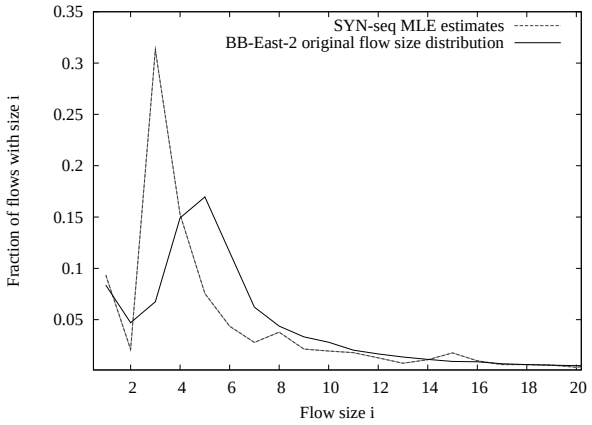

Figure 7: Estimated flow size distribution from the BB-East-2 trace versus the original flow size distribution. Packet sampling rate $p = 0.01$ using the SYN-seq MLE.

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 $\vec{\theta}$, $\tilde{\vec{\theta}}_{(1)}, \ldots, \tilde{\vec{\theta}}_{(m)}$, obtained at m monitors. It assumes all monitors will sample independently. Their goal is to find a new estimate $\tilde{\vec{\theta}}'$ of $\vec{\theta}$ using a linear combination of the local estimates $\tilde{\vec{\theta}}_{(1)}, \ldots, \tilde{\vec{\theta}}_{(m)}$ 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 information from combined samples. In this section, we 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.

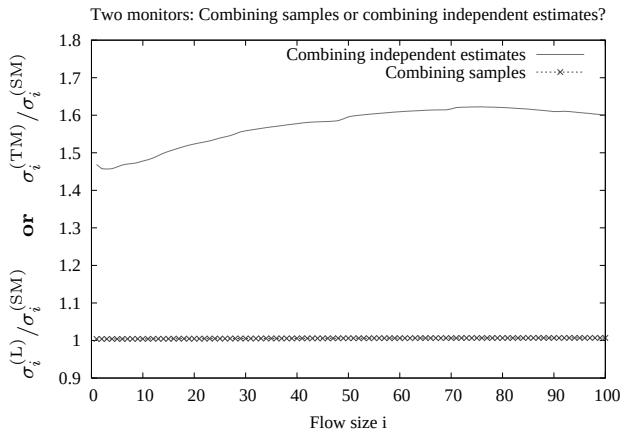

Figure 8: This graph uses the Cramér-Rao bound to show the advantage of making the estimation with the combine samples taken at each monitor over the combination of the two independent estimations taken at each monitor. The evaluation is done over distribution from trace BB-East-2.

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 8 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 of (2) and (3) in respect to (1). Let $\tilde{\vec{\theta}}'^{(\mathrm{L})}$ be the estimates obtained by the approach described in [4]; $\tilde{\vec{\theta}}^{(\mathrm{SM})}$ be the estimates obtained by the single monitor with sampling rate p; and $\tilde{\vec{\theta}}^{(\mathrm{TM})}$ 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 8 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 at one monitor with double the rate. The results also shows that combining two independent estimates increases the standard deviation error of the estimates in 50%. Using a central site comes with the cost of transferring all data to the central site. One can reduce this cost by sending summarized data.

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 types of information, such as 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 TCP sequence number information is essential for accurate flow size estimation. We also explored the benefit of including SYN flag information, and determined that the former is useful in reducing the errors associated with estimating the fraction of small sized flows. Using this as a starting point, we presented MLEs based on the conjugate gradients method, which come close to the Crámer-Rao bound, even for small sample set sizes. 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 made on observations at individual monitoring sites. To our knowledge this is the first study of flow size estimation from samples collected at multiple monitors.

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. And also the fruitful discussions with George D. Konidaris on MLEs. We also acknowledge the many anonymous reviewers for their comments and suggestions.

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.

9. 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 0́5: 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/ipfixcharter.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.