Optimal Skampling for the Flow Size Distribution

Darryl Veitch, Fellow, IEEE and Paul Tune, Member, IEEE

Abstract—We introduce a new method of data collection for flow size estimation, the Optimized Flow Sampling Sketch, which combines the optimal properties of Flow Sampling with the computational advantages of a counter array sketch. Using Fisher Information as a definitive basis of comparison, we show that the statistical efficiency of the method is within a constant factor of that of Flow Sampling, which is known to be optimal but which cannot be implemented without a flow-table, which has higher memory and computational costs. In the process we derive new results on the Fisher-Information theoretic and variance properties of the counter array sketch, proving that an overloaded sketch actually destroys information. We revisit the ‘Eviction Sketch’ of Ribeiro et alia using the Fisher Information framework. We show that its performance is much higher than previously supposed, and we define a new method, the Optimized Eviction Sketch, which has very high efficiency. We compare these methods against each other and a third skampling method, ‘Sketch Guided Sampling’, theoretically, on models and on data.

Index Terms—Cramér-Rao lower bounds, Internet, Maximum Likelihood Estimation, sampling methods, sketching.

I. INTRODUCTION

The distribution of flow size, that is the number of packets in a flow of packets, is an important metric for numerous networking applications including traffic modeling, management, and anomaly detection. In resource constrained environments such as within core Internet routers, limitations on processing time and memory make exact measurement of most metrics, including the flow size distribution, impossible. To address this problem, two approaches to fast approximate measurement have been traditionally taken: sampling and sketching. In this paper we present a hybrid skampling approach to the estimation of the flow size distribution.

We will assume that a flow key identifier can be calculated for each packet and that the first packet of the flow can be identified. The canonical example here is that of TCP connections, which begin with a SYN packet. Our work can also be applied in other contexts such as stream processing and ‘Big Data’, and discussed in the conclusion.

In traffic sampling, estimation is based on a rapidly chosen subset of the incoming packet stream. This is the dominant approach currently, for example it is used in Cisco’s Netflow [1], which is based on a simple pseudo-random per-packet sampling. Several works have detailed the significant estimation problems arising from such a scheme (e.g. [4], [8], [14]). An alternative approach, introduced in [6], is i.i.d. flow sampling (FS), where the random sampling decision is made directly on flows, so that either all packets in a given flow are sampled, or none.

In our prior work [8], [19], [20], we have shown that FS is statistically superior to all currently proposed alternatives, including Sample and Hold [5], and methods such as Dual Sampling [20] and others [17] exploiting TCP sequence number information[7]. This conclusion was based on a rigorous evaluation, in terms of Fisher Information, of the inherent abilities of the sampling methods as information collectors.

While the statistical performance of FS is excellent, its implementation carries inherent costs. Being based on the idea of indexing into a flow-table which is by definition collision-free, each entry in the table must store a unique flow key along with the packet counter, and crucially, this key must be read and compared on each sampled packet. In router measurement this is unsustainable in the worst case, namely where one or more flows with long bursts of short closely-spaced packets arrive, each of which requires a flow key lookup. In this paper we seek to avoid the memory, computation, and latency costs of flow-table based methods.

Sketches are compact data structures with fast update rules that are used to approximately measure properties over data streams. Sketches have been proposed to measure many metrics including set membership [2], entropy [25], heavy hitters [5], and flow sizes [9].

Although there have been a number of works combining sampling and sketching in some form to estimate various metrics [11], [12], [2], [25], [13], [16], [10], the traffic sampling and sketching worlds have remained largely separate, and key questions like which approach is statistically superior for which metrics, and how they might be optimally combined, are very much open.

This paper introduces a new data collection skampling scheme, so called as it combines both sampling and sketching features, called the OFSS or Optimized Flow Sampling Sketch. The FSS is a skampling method consisting of a front-end sampling component, namely FS, which feeds its output to a sketching component ‘Sk’, the counting array sketch of [9]. The OFSS is an FSS tuned according to an optimal tradeoff between the information gathering strengths and weaknesses of the FS and Sk components.

Statistically OFSS has FS-like performance and intuition, but as it is essentially a counter sketch computationally,

1 Caveats exist in special cases of no practical importance [20].
it has no flow-table costs and associated bottlenecks. We analyze the performance of the scheme in terms of Fisher Information, and compare it against FS and Sk. We show how the scheme can be tuned depending on which part of the distribution is deemed most important, and we give an algorithm allowing optimal parameter settings to be determined (an implementation is available on-line [23]). In designing OFSS we build on and significantly extend the Fisher analysis of Sk presented in [21]. In particular we give a proof of the key insight, that overloading Sk results in information destruction.

Our second main contribution is a Fisher Information analysis of an existing flow size estimation approach, due to Ribeiro et alia [16], [15], which we name the Eviction Sketch (ESk). This skampling method is based on an enhanced form of Sk where flows are split into L flow classes. Whereas [16] focuses on the use of L = 2, we show formally that much higher values provide much better performance. We define a form of ESk based on optimizing L as a function of flow load, the Optimized ESk (OESk), and compare it in detail against OFSS. We find that the methods are roughly comparable (see the conclusion for a summary), with respectable efficiencies compared to Flow Sampling, but that OFSS has a number of advantages linked to its greater simplicity.

Finally, we provide some comparisons, using both models and real traffic data, between the above methods. In the case of the data we define (also for the first time in most cases) Maximum Likelihood Estimators (MLEs) to perform the estimation based on the observables provided by the collection methods. We include in the comparisons another skampling method, the Sketch Guided Sampling (SGS) approach of Kumar et alia [11]. We provide the first Fisher Information analysis of SGS, which allows, as for the other methods, the evaluation of its Cramér-Rao lower bound on models, and its MLE on data. We find SGS to have extremely poor performance, and note that computationally it incurs the costs of both a sketch and a flow-table. We conclude that the method should not be considered further.

The work [9] of Kumar et alia, which introduced the counter sketch Sk, also describes a more sophisticated variant, a multi-resolution filter, which, effectively, contains a set of FSSEs as components. However there has been no formal analysis of this scheme in terms of Fisher Information, nor in terms of understanding and optimizing the sampling and sketching interactions. As far as we are aware, OFSS and OESk are the first data collection methods with an integrated design, incorporating both sampling and sketching, that address optimal information gathering directly. Our approach is also, with the exception of our earlier work [21], [22] (see also [15]), the first application of Fisher Information techniques to the analysis of sketching in any form. A short (6 page) version of this work, containing a subset of results on FSS/OFSS and a partial analysis of ESk, without proofs, appeared in [22].

The paper is organized as follows. Section II gives background on the Fisher Information framework, and defines and recalls key results for FS and Sk. Section III analyzes the performance of Sk in depth, and contains the results which are critical to the analysis of both OFSS and OESk. Section IV defines FSS and derives its Fisher Information and CRLB. Section V defines and analyzes OFSS, and explores the behavior of the optimal parameter as a function of flow load, and details how to calculate it. Section VI introduces ESk and provides its Fisher Information analysis, which Section VII then exploits to define OESk and describe its properties. Section VIII then formally compares FSS and ESk, and OFSS and OESk. Section IX introduces and analyzes SGS, and compares it to that of FS, OFSS and well calibrated ESk, using both models and traffic data. The derivations of the MLEs are left to the Appendix. We conclude in Section X.

II. FS, SK, AND FISHER INFORMATION

In this section we introduce the conceptual framework and provide required background on Fisher information and the associated Cramér-Rao lower bound. We define the two flow size summary methods: flow sampling (FS) and the counter sketch (Sk), that are central to the study of both FSS and ESk, and derive their Fisher Information. In the case of FS we recall the results from [20]. For Sk we follow the approach from [21] but offer a more complete (and corrected) treatment of the basic setup. Later in Section III we expand on the study of Sk considerably. Note that an initial analysis in terms of Fisher Information of Sk, with an emphasis on finite counter size, can be found in [15].

We write vectors (column by default) in bold lower-case, matrices in bold upper case, AT denotes the transpose of A, and diag(x) a m × m diagonal matrix whose diagonal entries are taken from the vector x ∈ Rm. We denote the index set {1, 2, . . . , m} by [m].

A. Modeling Framework

Consider a measurement interval of duration T containing Nf flows. As discussed in detail in [20], we can assume Nf is known. Of these flows, Mk have size k packets, 1 ≤ k ≤ W, where W < ∞ is the maximum flow size. There are n = W

\[ \sum_{k=1}^{W} kM_k \] packets in total and the average flow size is D = n/Nf.

Let θk = Mk/Nf. The flow size distribution, the unknown vector parameter we seek information on, is θ = [θ1, θ2, . . . , θW]T, and obeys

\[ 0 < θ_k < 1, \ k ∈ [W], \ \sum_{k=1}^{W} θ_k = 1. \] (1)

This is a deterministic model of the data over the measurement interval: randomness enters later through the action of the measurement method itself.

Any estimator of θ is based on an underlying observable which summaries the traffic. For each of the methods FS and Sk, this observable takes the form of a packet count C. This is a random variable, taking integer values j ≥ 0,
whose probability distribution \(c_j(\theta)\) depends on the details of the method as well as \(\theta\). Viewed as a function of \(\theta\) for a fixed value of \(j\) of the observed data, it is known as the likelihood: \(f(j, \theta) = c_j(\theta)\).

We use the unconditional formulation developed in [19], which includes \(j = 0\) sampled packets as a valid observation. This is possible since \(N_p\) is known, and it allows much simpler expressions for likelihoods. We also assume that \(N_p\) is large enough so that sets of observations are approximately independent and identically distributed (i.i.d.).

**B. Fisher Information and Constraints**

The Fisher information is a well known measure of the amount of information the observable holds about the unknown parameters, and is defined as

\[
J(\theta) = \mathbb{E}[\nabla_\theta \log f(j; \theta)](\nabla_\theta \log f(j; \theta)^T)
= \sum_{j \geq 0} (\nabla_\theta \log f(j; \theta))(\nabla_\theta \log f(j; \theta)^T)c_j. \tag{2}
\]

Recall that an \(n \times n\) real symmetric matrix \(A\) is positive (semi)definite if for all vectors \(z \in \mathbb{R}^n\setminus\{0\}\), \((z^T A z \geq 0)\). We write \(A \succeq 0\) when \(A\) is positive semidefinite. The Fisher information obeys \(J \succeq 0\), but not \(J > 0\) in general.

The great importance of the Fisher information lies in its connection to estimation variance. It is known that, if it is the Cramér–Rao Lower Bound (CRLB) exists, the estimator of \(\theta\) has the connection to estimation variance. It is known that, if it exists, the Cramér–Rao Lower Bound (CRLB) \(J^{-1}\) is the lower bound on the covariance matrix \(\Sigma_\theta\) of any unbiased estimator of \(\theta\), i.e. \(\Sigma_\theta \succeq J^{-1}\) in the positive semidefinite sense.

The above definition of \(J\) is in the familiar, unconstrained case. Here, however, \(\theta\) is subject to the constraints in \((1)\). Constraints provide additional knowledge of \(\theta\) without a single measurement being taken, and results in higher \(J\), and hence lower covariance. The constrained CRLB is [6]

\[
I^+ = J^{-1} - J^{-1} G (G^T J^{-1} G)^{-1} G^T J^{-1}
\]

where \(I^+\) denotes the Moore-Penrose pseudo-inverse [7] of the constrained Fisher information matrix \(J\). Here the gradient matrix is \(G(\theta) = \nabla_\theta (1_{W}^T \theta - 1) = 1_{W}\), where \(1_{W}\) is a \(W \times 1\) vector of ones, and so the above expression reduces to

\[
I^+ = J^{-1} - J^{-1} 1_{W} 1_{W}^T J^{-1} 1_{W} J^{-1} 1_{W}. \tag{3}
\]

With a single constraint, \(I^+\) has rank \(W - 1\) and so is singular.

An alternative expression for \(I^+\), which does not require invertibility of \(J\), due to Stoica and Ng [18], is given by

\[
I^+ = U(U^T J U)^{-1} U^T \tag{4}
\]

where \(U\) is a (non-unique) matrix whose columns are an orthonormal basis of the nullspace of \(G\). The quantity \(J_c = U^T J U\) carries an interpretation of the constrained \(J\) in the lower dimensional constrained space and is invertible under general conditions. For our problem the matrix \(U\) is \(W \times (W - 1)\) and \(J_c\) is of size \(W - 1\).

Of great importance here are the diagonal entries of \(I^+\), since \(\text{Var}(\hat{\theta}_k) \geq (I^+_{kk})^{-1}\) for any unbiased estimator. Comparison of these methods corresponds to comparing the best performance the schemes are capable of supporting, thereby reflecting their comparative efficiency in extracting information from the traffic stream.

By our independent sampling assumption, the Fisher information arising from the entire measurement interval is just that of a single counter \(C\) multiplied by the number of counters.

**C. Flow Sampling and its Fisher Information**

Conceptually FS is very simple: flows are sampled independently with probability \(p_f\) and dropped with probability \(q_f\). Sampling a given flow means that each packet within it is sampled, otherwise none of its packets are. Hence in this case the variable \(C\) represents the size of a typical (i.e. randomly selected over the measurement interval) sampled flow.

The (unconditional) distribution for the sampled flow size takes the simple form

\[
c_j = \sum_{k=1}^W b_{jk} \theta_k, \quad 0 \leq j \leq W, \tag{5}
\]

or \(c = B \theta\), where \(c = [c_0, c_1, c_2, \ldots, c_W]^T\). Here \(b_{jk}\) is simply the probability that if the original flow had \(k\) packets, only \(j\) remain after sampling. The sampling matrix \(B\) characterizes the sampling method.

For flow sampling the unconditional likelihood reduces to something particularly simple: for a flow of size \(k\), \(c_0 = q_f, c_k = p_f, \text{ and } c_i = 0 \text{ for all } i \neq \{0, k\}\). The corresponding sampling matrix is just

\[
B = \begin{bmatrix} q_f 1_W^T \\ p_f \text{diag}(1_W) \end{bmatrix}. \tag{6}
\]

It is not difficult to derive an explicit expression for \(J\):

\[
J_{FS} = p_f \text{diag}(\theta_1^{-1}, \theta_2^{-1}, \ldots, \theta_W^{-1}) + q_f 1_W 1_W^T, \tag{7}
\]

and from [19], \(I_{FS}^+ = J_{FS}^{-1} - \theta \theta^T = \frac{1}{p_f} \left( \text{diag}(\theta) - \theta \theta^T \right) \). The diagonal terms which bound the individual variances are then

\[
(I_{FS}^+_{kk}) = \frac{\theta_k(1 - \theta_k)}{p_f} \text{ for all } k \in [W]. \tag{8}
\]

Not surprisingly, for each \(k\) the variance bound is a decreasing function of \(p_f\).

The great strength of FS compared to other methods is that estimators based on it do not have to try to correct distortions: collected flows are perfect.
D. Counter Sketch and its Fisher Information

The counter sketch introduced by Kumar et alia [9] consists of an array of \( A \) packet counters, initialized to zero at the beginning of the measurement interval. Incoming packets are mapped independently and uniformly over the counters, using a hash function mapping a flow key, so that each packet in a flow maps to the same counter, but collisions can occur so that a given counter may sum the packet counts from two or more flows. We define \( \alpha = N_f/A \) to be the flow load factor, the average number of flows per counter. Here \( C \) is the final packet count in a typical counter, at the end of the measurement interval when all \( N_f \) flows are in the sketch.

From the i.i.d. nature of flow insertion, it is not hard to see that the counter value can be expressed as a sum of independent random variables \( \{kX_k\} \):

\[
C = \sum_{k=1}^{W} kX_k, \tag{9}
\]

where \( X_k \), which is binomially distributed with parameter \( (M, 1/A) \), gives the number of flows of size \( k \) in the counter, each of which contributes \( k \) packets to \( C \). However, \( C \) itself is far from binomial, and its density, and hence the likelihood, does not have a simple closed form.

To simplify calculations, we approximate the distribution of \( X_k \) as Poisson with parameter \( \lambda_k = M_k/A = \alpha \theta_k \), which is justified when \( N_f \) is large with non-vanishing load \( \alpha \). This yields a simple closed form expression for the generating function \( C^*(s) \) of \( C \), \( |s| < 1 \):

\[
C^*(s) = \prod_{k=1}^{W} e^{\lambda_k(s^{k-1})} = \exp\left(\sum_{k=1}^{W} \lambda_k(s^k - 1)\right). \tag{10}
\]

We now derive a recursion relation for the density \( c_j \) of \( C \).

**Theorem 1:** For \( j > 0 \) and \( 1 \leq k \leq W \)

\[
\frac{\partial c_j}{\partial \theta_k} = \begin{cases} 
\alpha c_{j-k} - \alpha c_j, & \text{if } j \geq k, \\
-\alpha c_j, & \text{otherwise.} 
\end{cases} \tag{11}
\]

**Proof:** Recalling that \( \lambda_k = \alpha \theta_k \), from \( \ref{eq:9} \) we have

\[
\frac{\partial C^*(s)}{\partial \theta_k} = \frac{\partial \lambda_k \partial C^*(s)}{\partial \lambda_k \partial \theta_k} = \alpha(s^k - 1)C^*(s) = \sum_{j \geq 0} \frac{\partial c_j}{\partial \theta_k} s^j.
\]

The result follows by writing \( C^*(s) = \sum_{j \geq 0} c_j s^j \) and equating coefficients of \( s^j \).

Note that we have not at this point imposed the constraint \( \sum \theta_k = 1 \), and so it is not (yet) true that \( \sum \lambda_k = \alpha \).

A general solution to this recursion relation is

\[
c_j = \left( \sum_{x \in \Omega_j} \prod_{k=1}^{W} \frac{\lambda_{x_k}}{x_k!} \right) e^{-\Lambda} \tag{12}
\]

where \( x_k \) is the sample value of \( X_k \) from \( \ref{eq:9} \). \( x = [x_1, x_2, \ldots, x_W] \) denotes a collision pattern, and \( \Lambda = \sum_{k=1}^{W} \lambda_k \). Here \( \Omega_j \) is the set of flow collision patterns with packet count \( C = \sum_{k=1}^{W} kx_k = j \). The first few density values are

\[
c_0 = e^{-\Lambda}, \quad c_1 = \lambda_1 e^{-\Lambda}, \quad c_2 = \left( \lambda_2 + \frac{\lambda_1^2}{2!} \right) e^{-\Lambda}, \quad c_3 = \left( \lambda_3 + \lambda_2 \lambda_1 + \frac{\lambda_1^3}{3!} \right) e^{-\Lambda}, \quad c_4 = \left( \lambda_4 + \lambda_3 \lambda_1 + \frac{\lambda_2^2}{2!} + \lambda_2 \lambda_1^2 + \frac{\lambda_1^4}{4!} \right) e^{-\Lambda}.
\]

Using \( \ref{eq:11} \) in the definition of Fisher information, it is not difficult to show that

\[
(J)_{ik} = \alpha^2 \left( -1 + E \left[ \frac{c_{j-i} c_{j-k}}{c_j} \right] \right \}_{j \geq \max(i,k)} \tag{13}
\]

or in matrix form \( J(\theta) = \alpha^2 \left( \sum_{j=1}^{\infty} \frac{1}{c_j} z_j z_j^T - 1_W 1_W^T \right) \),

where \( z_j = [c_{j-1}, \ldots, c_0, 0, \ldots, 0]^T \), a \( W \times 1 \) vector with zero entries from \( (j+1) \)-th position onwards if \( j < W \).

To compare across methods, it is convenient to renormalize to the per-flow view used by FS. Thus we define the per-flow Fisher Information for \( S_k \) as

\[
J_{S_k} = A J/N_f = J/\alpha = \alpha \left( \sum_{j=1}^{1} \frac{1}{c_j} z_j z_j^T - 1_W 1_W^T \right). \tag{15}
\]

The constrained per-flow CRLB, \( I^+ \), is given by substituting \( J_{S_k} \) into \( \ref{eq:15} \). In what follows we adhere to this convention that the per-flow Fisher Information and CRLB carry method subscript labels, whereas the per-counter quantities do not.

To help keep track of key notations we provide Table I.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( W )</td>
<td>Maximum flow size (in packets)</td>
</tr>
<tr>
<td>( N_f )</td>
<td>Number of flows in measurement interval</td>
</tr>
<tr>
<td>( M_k )</td>
<td>Number of flows size ( k )</td>
</tr>
<tr>
<td>( D )</td>
<td>Average flow size (in packets)</td>
</tr>
<tr>
<td>( n )</td>
<td>Total number of packets</td>
</tr>
<tr>
<td>( \theta_k )</td>
<td>Proportion of flows of size ( k ) ( (\theta_k = M_k/N_f) )</td>
</tr>
<tr>
<td>( \theta )</td>
<td>Flow size distribution ( {\theta_k, k \in [W]} )</td>
</tr>
<tr>
<td>( C )</td>
<td>Counter random variable</td>
</tr>
<tr>
<td>( c_j )</td>
<td>Discrete density of ( C ): ( c_j = Pr{C = j} )</td>
</tr>
<tr>
<td>( p_f )</td>
<td>Flow sampling probability</td>
</tr>
<tr>
<td>( B )</td>
<td>Sampling matrix of sampling methods</td>
</tr>
<tr>
<td>( A )</td>
<td>Number of packet counters in sketch</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>Flow load factor (per counter)</td>
</tr>
<tr>
<td>( \lambda_k )</td>
<td>Rate parameter of flow size ( k ): ( \lambda_k = \alpha \theta_k )</td>
</tr>
<tr>
<td>( J )</td>
<td>Unconstrained Fisher information of ( \theta )</td>
</tr>
<tr>
<td>( I )</td>
<td>Constrained Fisher information of ( \theta )</td>
</tr>
<tr>
<td>( I^+ )</td>
<td>Pseudo inverse of ( I ) (CRLB of ( \theta ))</td>
</tr>
</tbody>
</table>

**TABLE I**

**KEY NOTATIONS.**
III. The $\alpha$ Dependence of $S_k$

We now examine the dependence of $J_{S_k}(\alpha)$ and $T_{S_k}^+(\alpha)$ on the per-count flow load $\alpha$, as these will be central below.

A. Monotonicity of $J_{S_k}(\alpha)$ and $T_{S_k}^+(\alpha)$

For general $\alpha$ it is difficult to obtain simplified expressions for $J_{S_k}$, much less $J_{S_k}^-$ or $T_{S_k}^+$. Nonetheless, it is possible to show that each have monotonic behavior in $\alpha$. We will need the following lemma. This result was given in [20] in the more restrictive context of sampling methods. Here we give a general proof based on the representation $T^+ = U(U^T J U)^{-1} U^T$ (Equation (4)) for general $G$ of dimension $m \leq n$, and any $J$ provided $(U^T J U)^{-1}$ exists.

**Lemma 2:** Assume $J_1, J_2$ are unconstrained $n \times n$ Fisher Information matrices such that $U^T J_i U$ is invertible, $i = 1, 2$. Then $J_2 < J_1$ if and only if $T_{J_2}^+ > T_{J_1}^+$. In addition $J_2 < J_1$ if and only if $T_{J_2}^+ > T_{J_1}^+$ when restricted to the nullspace of $G$.

**Proof:** By definition $v^T J_2 v < v^T J_1 v$ for any $v \in R^n$. Thus for any $x \in R^{n-m}$, we can set $v = Ux$ and write $x^T U^T J_i U x \leq x^T U^T J_i U x$. Since $x$ is a general vector in $R^{n-m}$, we have shown that $U^T J_i U \leq U^T J_i U$, and hence, by a standard result on positive semidefinite matrices, $(U^T J_i U)^{-1} \geq (U^T J_i U)^{-1}$.

Now take any $v \in R^n$, and define $x = U^T v \in R^{n-m}$, that is $x$ is a general vector in the nullspace of $G$. Thus $v^T U(U^T J_i U)^{-1} U^T v \geq v^T U(U^T J_i U)^{-1} U^T v$, or in other words $T_{J_2}^+ \geq T_{J_1}^+$ for vectors in the nullspace of $G$, whereas $T_{J_2}^+ = T_{J_1}^+$ otherwise, hence $T_{J_2}^+ > T_{J_1}^+$ in the general case. The result for the strict inequalities follows in the same way by simply omitting the last step.

Note that the invertibility of $U^T J_i U$ is a necessary condition for the existence of unbiased estimators obeying the constraint [18], so that the condition of Lemma 2 is the most general possible.

**Theorem 3:** The matrix $J_{S_k}$ is decreasing in the positive semidefinite sense. In addition $T_{S_k}^+$ is increasing when restricted to the nullspace of $G$, and $(T_{S_k}^+)^{kk}$ increasing for each $k$.

**Proof:** Since $J$ is a function of $A$ and $N_j$ only via $\alpha = N_j/A$, we are free to consider $\alpha$ increasing from zero via $A$ decreasing from infinity with $N_j$ fixed. For example by a suitable choice of $N_j$ for any initial arbitrarily small $\alpha$, $A$ can be assumed to be a power of 2, and counters can be merged pairwise, with counter values added, resulting in $A \to A/2$ and $\alpha \to 2\alpha$ at each step (the identity of the counters merged does not matter due to their independence). The amount of information input to the sketch, and the number of flows, is constant, but clearly the per-flow information contained in the sketch drops monotonically at each step: $J_{S_k}(\alpha_2) \leq J_{S_k}(\alpha_1)$ for $\alpha_2 = 2\alpha_1 > \alpha_1$, since

$$J_{S_k}(\alpha_2) = \sum_{i=1,3,5,...,A-1} \mathbf{J}(C_i + C_{i+1}) \leq \sum_{i=1,3,5,...,A-1} \mathbf{J}(C_i) = J_0 \left(C_1, C_2, \ldots, C_A \right) = J_{S_k}(\alpha_1)$$

where the second line follows from the Data Processing Inequality for Fisher Information [21]. Equality holds if and only if $C_{12} = C_1 + C_2$ is a sufficient statistic for $\theta$ based on $(C_1, C_2)$.

**B. Asymptotic behavior of $J_{S_k}(\alpha)$**

Denote by $\mu_{ij}$ the $n$-th centered (formal) moment of $\theta$. It is easy to show that the mean and variance of $C$ are given by $\mu = \alpha \sum_j j \beta_j = \alpha \mu_1'$, and $\sigma^2 = \alpha \sum_j j^2 \beta_j = \alpha \mu_2'$ respectively. Note that these each diverge with $\alpha$. Since $C$ is a sum of independent random variables with finite variances, the Central Limit Theorem can be used to justify approximating the counter distribution as $C \sim N(\mu, \sigma^2)$.

**Theorem 4:** The large-$\alpha$ asymptotic form of $J_{S_k}$ is

$$J_{S_k}(\alpha) = \alpha \left( e^{ik/\sigma^2 - 1} \right) = \sum_{j=1}^{\infty} \alpha^{1-j} \frac{1}{j!} (\frac{ik}{\mu_2'})^j$$

where

$$\alpha = (\frac{ik}{\mu_2'})^2 + \frac{1}{\alpha^2} \left( \frac{ik}{\mu_2'} \right)^2 + \frac{1}{\alpha^2 3!} \left( \frac{ik}{\mu_2'} \right)^3 + \ldots$$

**Proof:** Taking the density $c_j$ as Gaussian, the integrand $(c_j-c_j-\kappa)/\sigma_j^2$ from [13] readily simplifies to $A \exp \left( \frac{b+c}{\sigma^2} \right)$, where $A = \exp \left( \frac{-\sigma^2}{2} \right)$ is independent of $j$ (we may ignore the indicator function as $\sum_j c_j$ vanishes at large $\alpha$). It follows that $E[(c_j-c_j-\kappa)/\sigma_j^2] = AM(\frac{b+c}{\sigma})$, where $M(t) = \exp(\mu t + \sigma^2 t^2/2)$ is the moment generating function of $N(\mu, \sigma^2)$. After a little algebra this simplifies to $E[(c_j-c_j-\kappa)/\sigma_j^2] = \exp(ik/\sigma^2)$. The result then follows from [15] and [13].

Note that if we had have modeled $C$ as a Gaussian from the outset, then the associated Fisher information would have been the first two terms of (17) only. This is a rank-2 matrix for all $W$, reflecting the fact that $\theta$ is unidentifiable for $W > 2$ if it is only known via its first two moments. In contrast, (17) results from using the Fisher Information from the true $C$, but evaluating it in an asymptotic regime where $c_j$ is close to Gaussian. This approach preserves the
full large-\(\alpha\) structure of \(J_{Sk}\), and like the true \(J_{Sk}\) is full-rank for all finite \(\alpha\).

From (17), \(J_{Sk}\) can be written as a matrix sum \(J_{Sk} = \sum_{j=1}^{\infty} \alpha^{-j} a_j D_j\), where \(a_j = (j! \mu_j)^{-1}\), and \(j\) indexes the \(\alpha\) dependence from the leading term of \(\alpha^0\) at \(j = 1\) down. Each \(D_j\) is rank-1 as it can be written as \(D_j = d_j d_j^T\), where \(d_j^T = [1, 2, 3, \ldots, W]^T\). Thus row \(i\) of \(J_{Sk}\) is \(r_i = \sum_{j=1}^{\infty} \alpha^{-j} a_j j^i d_j^T\). It is easy to see that \(\{d_j, j = 1, 2, \ldots, W\}\) are linearly independent, so \(J_{Sk}\) is full-rank. We can now prove the following.

**Theorem 5:** The asymptotic \(J_{Sk}\) given by (17) has positive eigenvalues having distinct leading term \(\alpha\) dependencies of \(\gamma_i \sim \alpha^{-1}, i \in [W - 1]\).

**Proof:** We first transform \(J_{Sk}\) to control the \(\alpha\) dependence of each row. Since \(a_1 D_1 = \mu_1 d_1 d_1^T\) is the matrix carrying the leading dependence, namely that of \(\alpha^0\), we can remove that dependence from each row and hence each element of \(J_{Sk}\) below row-1 by subtracting a multiple of \(r_1\) corresponding to it. That is \(r_i = r_i - \mu_1 r_1\), resulting in \(r_i = \sum_{j=2}^{\infty} \alpha^{-j} a_j j^i (j^i - i)\), for all \(i \geq 2\), which has a leading power of at most \(\alpha^{-1}\). Since the elements of each \(d_j\) are positive, and since \((j^i - i) > 0\) for all \(i \geq 2\), \(j \geq 2\), each row the coefficient of each power of \(\alpha\) is non-zero, so the new leading power in each element of \(J_{Sk}\) below row-1 is \(\alpha^{-1}\). The procedure can therefore be repeated beginning at row-2 to reduce the leading power to \(\alpha^{-2}\) in each element below row-2, since \(j^2 - i^2 - i > 0\) for all \(i \geq 3\), \(j \geq 3\). Because \(j^i - i^k - 1 - 1^k - 2 - \ldots - i > 0\) for \(i \geq k\), \(j \geq k\) for each \(k \geq 2\), the procedure can be extended to the last row, resulting in a \(J_{Sk}\) of positive elements where each element in the \(i\)-th row is \(\sim \alpha^{-i}\).

The next step is to reduce to echelon form. Beginning with using row-1 to clear column-1, it is easy to see that the leading power of \(\alpha\) across each row is unaffected, since to clear an element the higher alpha dependence of row-1 must already have been factored out to match that of the element, and hence of each element in that row. The same applies recursively, resulting in an upper diagonal matrix where the non-zero elements of the \(i\)-th row are each \(\alpha^{-i}\). Back substitution then yields a diagonal matrix with the same property. Since eigenvalues are invariant under diagonalization and the eigenvalues of a diagonal matrix are just the diagonal elements, the result follows.

**C. Asymptotic behavior of \(J_{Sk}^{-1}(\alpha)\) and \(\mathcal{I}^+(\alpha)\)**

Since the eigenvalues of \(J_{Sk}\) go as \(\{\alpha^{j-1}, 1 \leq j \leq W\}\), those of \(J_{Sk}^{-1}\) go as \(\{\alpha^{-1}\}\). The variance of the quantity \(\sum_{k=1}^{\infty} \alpha^k \theta_k\), associated to a generic weight vector \(\alpha = [a_1, \ldots, a_W]^T\) which has components in each eigenspace, is dominated by the largest eigenvalue and therefore goes as \(\alpha^{W-1}\). Intuitively, we might expect the constraint, which effectively reduces the dimension of the parameter space by \(1\), to improve the variance to \(\alpha^{W-2}\). The following result confirms this.

**Theorem 6:** For large \(\alpha\) the constrained CRLB \(\mathcal{I}^+(\alpha)\) has positive eigenvalues having distinct leading term \(\alpha\) dependencies of \(\gamma_i \sim \alpha^{-1}, i \in [W - 1]\).

**Proof:** We use the expression (4) for the pseudo inverse. Recall that \(J_c = U^T J_{Sk} U\). The key observation is that \(U^T D_j U = U^T d_j d_j^T U = v_j v_j^T\), where \(v_j = U^T d_j\) is a \(W \times 1\) vector, and so is rank-1. Since \(U^T U = U U^T\), the \(v_j\) inherit the linear independence properties of the \(d_j\). It follows that there exists a matrix expansion for \(J_c\) in terms of rank-1 matrices which mirrors that for \(J_{Sk}\). Theorem 3 now applies to \(J_c\), but with \(i\) restricted to \(1 \leq i \leq W - 1\) since \(J_c\) is of size \(W - 1\). The eigenvalues of \(J_c^{-1}\) therefore go as \(\{\alpha^{-1}, 1 \leq j \leq W - 1\}\). The final step \(\mathcal{I}^+_{Sk} = U J_c^{-1} U^T\) is just an embedding into \(R^W\), and so adds a new eigenvalue of 0, corresponding to the constraint direction \(G\), without altering the existing ones.

**D. The Information Carrying Capacity of \(Sk\)**

As \(\alpha\) increases the sketch fills, and at high values where flow collisions are severe, intuitively we expect that it will become increasingly difficult and ultimately impossible to invert for \(\theta\). In other words, we expect that overloading a sketch does not merely result in diminishing returns for information gathering, but that information is actually destroyed, so that in the limit nothing about \(\theta\) is known beyond that expressed by the constraints \(0 < \theta_k < 1\) and \(\sum_k \theta_k = 1\).

Since the total information in the sketch is, up to a factor \(A\), equal to the per-counter information, to examine this question definitively we examine the per-counter information and CRLB, namely (recall (15))

\[
J = \alpha J_{Sk}(\alpha),
\]

\[
\mathcal{I}^+ = \frac{\mathcal{I}^+_{Sk}(\alpha)}{\alpha}.
\]

First consider the information viewpoint. Although the per-flow \(J_{Sk}(\alpha)\) is monotonic decreasing (in fact to zero), indicating that the information carried by individual flows is being increasingly poorly preserved as expected, in fact asymptotically each element of the total information \(J\) is roughly proportional to \(\alpha\), which seems to suggest that the sketch is still storing information at the rate it is entering. The resolution to this apparent paradox lies in the fact that \(J\) has a spectrum of eigenvalues, each with its own \(\alpha\) dependency, corresponding to information being stored in different eigenspaces at different rates.

From Theorem 5 the dominant eigenspace of \(J\) has growth rate \(\alpha\) and corresponds asymptotically to the 'mean direction' of \(a = [1, 2, 3, \ldots, W]^T\). To understand this, note that in the high \(\alpha\) limit each counter value will be tightly clustered around \(\alpha D = \alpha \sum_{k=1}^{W-1} k \theta_k = \alpha \alpha^T \theta\), so the mean \(D\) is known with increasing precision as \(\alpha\) increases, and the corresponding variance goes as \(\alpha^{-1}\). However in the \(\mathcal{I}^+\) world the dominant eigenspace is at the opposite end of the spectrum with a growth rate of \(\alpha^{W-3}\) (Theorem 6), and this is the asymptotic variance rate that will be seen by a generic direction there.

Now consider Equation (19). It is a product of two terms, each a function of \(\alpha\). The first, \(1/\alpha\), corresponds to the
information input to the sketch and is a monotonically decreasing scalar. The second, \( T_{Sk}(\alpha) \), is a per-flow covariance matrix which is monotonically increasing according to Theorem 5 (the higher the load, the worse Sk’s per-flow performance). We expect that this competition will result in a global minimum for generic directions \( a \) because of the extreme ambiguity of the sketch (reported in [21]), which will make \( a^TT_{Sk}(\alpha)a \) grow much faster than linearly.

To show this in the context of the individual variances, we define

\[
I_k(x) = \frac{e_k^TT_{Sk}(x)e_k}{x} = (T_{Sk}(x))_{kk},
\]

where \( e_k(k) = 1, \) zero elsewhere.

**Theorem 7:** For each \( k, \) \( I_k(x) \) has a global minimum at finite \( x = \alpha_k^* > 0 \) for any \( \theta \) with \( W > 3. \)

**Proof:** At small \( \alpha, \) Sk(\( \alpha \)) defaults to FS(1), so from [5] we have \( I_k(x) \approx \theta_k(1 - \theta_k)/x, \) which decreases from \( \infty \) inversely as load \( x \) increases above zero. On the other hand, we know from Theorem 6 that the variance bound for generic directions, and hence \( I_k(x) \) since \( e_k \) is not orthogonal to the dominant eigenvector of \( T_{Sk} \) for any \( k, \) will have the scaling \( x^{W-3} \) of the dominant eigenvector, and hence diverge with \( x \) provided \( W > 3. \) It follows that for each \( k \) when \( W > 3 \) there exists a unique global minimum at some finite \( x > 0. \)

The significance of Theorem 7 is that it confirms the intuition that there is a maximum amount of total information that can be stored in the sketch. If more information is available externally (higher \( \alpha \)), not only can it not be stored in the sketch, attempting to do so results in degradation and loss of information.

Note that the above result does not necessarily hold when \( e_k \) is replaced with a direction which aligns with any of the lesser eigenvectors of \( T_{Sk}(\alpha). \) In particular, the variance bound is of course zero in the constraint direction \( a = [1, 1, \ldots, 1]^T, \) and as already noted, in the direction \( a = [1, 2, 3, \ldots, W]^T \) corresponding to the mean of \( \theta, \) the variance decays to zero with increasing \( \alpha, \) and so there is no global minimum at finite \( \alpha \) for any \( W. \)

The case \( W = 2 \) is special. Here, the combination of the constraint \( 1 = \theta_1 + \theta_2, \) and the certain knowledge of \( D = \theta_1 + 2\theta_2, \) allows \( \theta \) to be exactly determined in the limit, which is reflected in the variance of even the generic (the worst) direction dropping as \( \alpha^{-1}. \)

IV. FSS

In this section we define the Flow Sampling Sketch (FSS), a family of skampling methods indexed by an internal parameter \( p_f, \) the flow sampling probability, then describe its properties and derive its Fisher Information.

A. Definition

To motivate the definition of skampling methods, we begin by contrasting the nature of FS and Sk.

The central issue is that of collision resolution. Allowing multiple flows to map to the same counter, that is to collide, is the origin of the high speed and low memory implementation advantages of Sk, and sketches in general. It is also the origin of their statistical weakness: collisions create complex ambiguities which literally destroy information about the flow size distribution, and oblige estimators to attempt an error-fraught non-linear inversion from the counter values back to \( \theta. \) Sampling methods also destroy information and can lead to complex inversion [20], though the nature of the induced ambiguity is very different [21].

The key advantage of FS is that it is the exception to this rule: each sampled flow is undistorted. The information loss in FS is due solely to the fact that only a subset of flows are sampled, and if \( p_f = 1, \) there is no loss at all. However, as discussed earlier, the requirement of strict collision resolution makes FS, for any \( p_f, \) more resource intensive.

In light of the above, we set the following objective for an effective skampling method for \( \theta: \) to achieve FS-like performance and properties, in an implementation allowing collisions. To achieve this goal the inevitable collision-related ambiguities must be somehow controlled for any given counter array size \( A \) (and hence limited memory budget).

The inspiration for FSS arises from the finding from Theorem 7 that load levels in Sk past a certain value result in information destruction: thus if \( \alpha \) could be kept low, then the counter array could work at an ‘operating point’ with low information loss. A natural way to achieve this is to perform a thinning of the incoming flows to reduce the input \( \alpha \) to an effective \( \alpha' < \alpha \) seen by the sketch, set to a suitable value, related to the global minimum of Theorem 7 which trades off the need for low ambiguity (\( \alpha' \) not too big), and adequate data collection as well as not under-utilizing expensive memory (\( \alpha' \) not too small).

**Definition:** FSS\((p_f; A)\) Each flow is sampled with probability \( p_f \). Packets of sampled flows are inserted into a counter sketch with \( A \) counters.

It is clear that FSS operates precisely as if an FS front end filtered flows before passing them to an unmodified counter sketch, which sees a reduced effective flow load of

\[
\alpha' = p_f \alpha.
\]

In terms of information, FSS combines the loss of information arising from FS due to only some of the flows being sampled, with an ambiguity loss due to the counter sketch for those flows stored within it.

A key advantage of FSS lies in how the FS component can be implemented. Since sampled flows are passed to the sketch, there is no need for any flow-table or collision resolution – the implementation issues of a true FS method are therefore completely bypassed. Only the flow selection itself remains, but this can be easily implemented in deterministic time on a per-packet basis using a hash function operating on flow keys. As described in [21], this

---

2It is useful to keep in mind a nominal value of \( \alpha' = 1, \) in which case \( 1 - c_0 = 1 - e^{-1} \approx 0.632 \) of the counters are used.
hash function can be simultaneously used to index into the sketch. It follows that FSS is just as implementable as Sk. (Note however that separate hash functions are preferable to obtain greater independence between the flow selection and flow summary stages.) It also follows, for \( p_f \) values such that \( \alpha' \) is small enough, that FSS can indeed be viewed as an implementable form of Flow Sampling.

As an FS-like method, FSS inherits a number of subsidiary advantages of FS, despite the fact that the data structure is a sketch. One of these is greater flexibility in adapting to varying traffic levels. In Sk if \( N_f \) increases suddenly then \( \alpha \) over that interval will become large, destroying almost all information about \( \theta \). The only possible response for a given memory budget is to reduce the duration \( T \) of subsequent measurement intervals. In FSS there is a second option: \( p_f \) could be simply decreased to keep \( \alpha' \) in a target range. This degree of freedom provides a decoupling of the time-scale over which traffic statistics are being collected, from the effectiveness of the data collection.

### B. Fisher Information

Whereas the number of flows \( N_f \) entering FSS is a known constant, the number \( N_f' \) reaching the counter array is a random variable. This complicates the calculation of the density of a counter variable \( C \).

Equation (29) still applies, but now \( X_k \) is distributed as Bin\((M_k, 1/A)\) where \( M_k' \sim \text{Bin}(M_k, p_f) \) is itself random. Remarkably, it turns out that \( X_k \) is still Binomial with parameter \( (M_k, p_f/A) \), so that the calculation of \( C'^*(s) \) proceeds as before with modified coefficients, yielding

\[
C'^*(s) = \prod_{k=1}^{W} e^{\lambda'_k (s^k - 1)} = e^{\sum_{k=1}^{W} \lambda'_k s^k}
\]

(21)

where \( \lambda'_k = \alpha' \theta_k = p_f \lambda_k \). A proof using generating functions is given in the Appendix.

It follows from the above that the density of \( C \) takes the same form as for Sk, with \( \alpha \) replaced by \( \alpha' \). Since the dependence on \( \theta \) is unchanged, it is easy to show that the per-counter information for FSS is just as for Sk with parameter \( \alpha' \), and therefore that the information per-incoming-flow is

\[
J_{\text{FSS}}(p_f, \alpha) = p_f J_{\text{Sk}}(\alpha')
\]

(22)

with corresponding CRLB

\[
\mathcal{I}^+_{\text{FSS}}(p_f, \alpha) = \frac{1}{p_f} \mathcal{I}^+_{\text{Sk}}(\alpha').
\]

(23)

Here the \( p_f \) factors arise because the per-flow normalization is with respect to the number \( N_f' \) of original incoming flows, whereas the number \( N_f' \) reaching the sketch is on average \( p_f N_f \).

By definition \( \mathcal{I}^+_{\text{FSS}}(1, \alpha) = \mathcal{I}^+_{\text{Sk}}(\alpha) \). Moreover, as \( \alpha \to 0 \) at fixed \( p_f \), clearly \( \mathcal{I}^+_{\text{FSS}}(p_f, \alpha) \to \mathcal{I}^+_{\text{FS}}(p_f) \), since each sampled flow has a separate record. Finally, it is intuitively clear that FSS with parameter \( p_f \) cannot gather more information than FS with the same \( p_f \), and so for a fixed \( p_f \), flow sampling outperforms FSS, as we now prove.

**Theorem 8:** For fixed \( p_f \), \( \mathcal{I}^+_{\text{FS}} \leq \mathcal{I}^+_{\text{FSS}} \) for any \( \alpha' > 0 \).

**Proof:** Denote by \( X \) the set of sampled flows, and by \( C \) the vector of counter variables. Clearly, these form a Markov chain \( \theta \to X \to C \). A straightforward application of the Data Processing Inequality for Fisher Information \([24]\) yields \( J_{\text{FS}} \geq J_{\text{FSS}} \). Since \( \alpha' > 0 \) implies both \( p_f > 0 \) and \( \alpha > 0 \), \( J_{\text{FS}} \), \( J_{\text{FSS}} \) are positive definite. It now follows from lemma \([2] \) that \( \mathcal{I}^+_{\text{FS}} \leq \mathcal{I}^+_{\text{FSS}} \).

From \([23]\), for fixed \( p_f \) only a fixed degree of flow thinning is provided by FSS, and as \( \alpha \) grows, the scheme will suffer information destruction from overloaded counters just as for Sk.

### C. Impact of the Number of Missing Flows

The sketch captures information on those flows within FSS which are sampled. Another potential source of information is the number \( M = N_f - N_f' \) of unsampled flows. The value of this random variable can be observed by using a simple additional counter which measures the number of flows. The value of \( N_f' \) can be achieved at high speed and low cost in the same way that \( N_f \) is measured (in the case of TCP, by incrementing a counter for each SYN packet). We now evaluate the increase in information due to observing \( M \).

The distribution of \( M = \text{Bin}(N_f, q_f) \), which we will again approximate as a Poisson variable with density \( f_m(m) = \text{Pr}(M = m) = e^{-\mu} \mu^m / m! \) where \( \mu = q_f N_f' \). By definition, \( N_f = \sum_{k=1}^{W} M_k = \sum_{k=1}^{W} N_f \theta_k = N_f \sum_{k=1}^{W} \theta_k \), and so \( \mu \) is an implicit function of \( \theta \), with \( \partial \mu / \partial \theta_k = q_f N_f' = \mu \) a constant for all \( k \). It follows (see Appendix for details) that the (per-flow) Fisher Information of observing \( M \) is \( J_M = q_f 1_W 1_T^T \).

Finally, if we assume that \( M \) is independent of the counter variables, we can write the net information available from FSS per-flow as

\[
J_{\text{FSS}}(p_f, \alpha) = p_f J_{\text{Sk}}(\alpha') + q_f 1_W 1_T^T
\]

(24)

To calculate the CRLB, \( \mathcal{I}^+_{\text{FSS}} \), we must first calculate \( \mathcal{J}^{-1}_{\text{FSS}} \). The matrix inversion lemma \([7]\) states that

\[
(R + STU)^{-1} = R^{-1} - R^{-1} S(T^{-1} + UR^{-1} S)^{-1} UR^{-1}.
\]

Applying this to \([24]\) with \( R = \alpha' J_{\text{Sk}}(\alpha') \), \( S = 1_W \), \( T = 1 \), and \( U = (\alpha - \alpha') 1_W^T \), we obtain

\[
\mathcal{J}^{-1}_{\text{FSS}}(\alpha) = \frac{\alpha'}{\alpha'} J_{\text{Sk}}(\alpha') - \frac{\alpha - \alpha'}{(\alpha' + (\alpha - \alpha') 1_W 1_T^T)^{-1} (\alpha' J_{\text{Sk}}(\alpha') 1_W 1_T^T)}
\]

with \( K' = 1_W 1_T^T (\alpha' J_{\text{Sk}}(\alpha') 1_W 1_T^T) \).

To calculate \( \mathcal{I}^+ \) we first post-multiply by \( 1_W \) to obtain

\[
K \equiv J_{\text{FSS}}(\alpha) 1_W = \frac{\alpha}{\alpha'} \left( 1 - \frac{(\alpha - \alpha') 1_W^T K'}{\alpha' + (\alpha - \alpha') 1_W 1_T^T} \right) \equiv c K',
\]

where \( c \) is a scalar. It follows that the quantity \( K = J_{\text{FSS}}(\alpha) 1_W \) for FSS differs from the analogous one \( K' \) for
Sk only by a scale factor, and similarly $B = 1^T_0 K = cB'$. Let $d = \frac{(\alpha - \alpha')}{(\alpha + (\alpha - \alpha'))B}$, so that $e = \frac{\alpha}{\alpha'}(1 - d)$. It is now straightforward to evaluate $\mathcal{I}_{FSS}^+$. From (3)

$$\mathcal{I}_{FSS}^+(\eta, \alpha) = J_{FSS}^1(\eta, \alpha) = \frac{KK^T}{B} = \frac{\alpha}{\alpha'} \left( J_{Sk}(\alpha') - \frac{d}{B'} \frac{K'K^T}{cB'} \right) = \frac{\alpha}{\alpha'} \left( J_{Sk}(\alpha') - \frac{K'K^T}{B} \right) = \frac{\alpha}{\alpha'} \mathcal{I}_{Sk}^+(\alpha') = \frac{1}{\eta} \mathcal{I}_{Sk}^+(\alpha'). \tag{25}$$

This result is exactly the same as (23, that is the inclusion of $M$, although modifying $J_{FSS}$ has not influenced the final CRLB. One way to understand this perhaps surprising result is to note that $J_M$ arose from rewriting $N_f$ as $N_f \sum_{k=1}^{W} \theta_k$, without which one would have had $J_M = 0$. The constrained inverse suppresses the contribution arising from this rewriting, effectively removing the information that $M$ added.

V. OFSS

In this section we introduce the Optimized Flow Sampling Sketch (OFSS), which is just FSS with the value of $\eta$ chosen to maximize its Fisher Information gathering ability. We explore the behavior of OFSS as a function of the external parameters $\theta$, $\alpha$, examine its efficiency, and explain how to calibrate it.

A. Definition

The parameters considered fixed in a given measurement run are the counter array size $A$, the number of flows $N_f$ in the measurement interval, and $\theta$. The input load to the FSS is therefore fixed at $\alpha = N_f/A$, whereas $\eta$ is an internal parameter free to be tuned to any value in $[0, 1]$.

In defining OFSS our aim is to select $\eta$ to maximize the total amount of information captured by FSS, or equivalently, to minimize the CRLB when using all $A$ counters. As before, since $A$ enters only as a multiplicative factor, we equivalently consider the per-counter quantities, namely

$$J(\eta, \alpha) = \eta J_{FSS}(\alpha) = \alpha J_{Sk}(\alpha'), \tag{26}$$

$$\mathcal{I}(\eta, \alpha) = \eta \mathcal{I}_{FSS}(\alpha) = \frac{\alpha}{\alpha'} \mathcal{I}_{Sk}(\alpha') = \frac{\alpha}{\alpha'} \mathcal{I}_{Sk}^+(\alpha'), \tag{27}$$

where $\eta$ appears on the right hand side only via $\alpha' = \eta \alpha'$.\footnote{Note that the total information $\mathcal{I}(\eta, \alpha)/A = \mathcal{I}_{Sk}^+(\alpha')/N_f$, which is simply the total covariance of $S_k$ storing $N_f$ flows with load factor $\alpha'$.}

We proceed as follows. The existence of the global minimum $\alpha'_{k*}$ of $I_k$ implies in particular the existence of a first local minimum $\alpha'_{k*} > 0$ obeying $\alpha'_{k*} \leq \alpha'_{k**}$. We base our definition of OFSS on $\alpha'_{k*}$ rather than $\alpha'_{k*}$ for reasons we give presently. Apart from the trivial case of $W = 2$, we do not know of any examples where the first local minima is not unique, and hence equal to the global minimum. We conjecture that $\alpha'_{k*} = \alpha'_{k**}$ in all cases with $W > 3$, however we were unable to prove this. Certainly for large $W$ values (1000 or more) typical in the network measurement context, the asymptotic $\alpha^{W-3}$ divergence will be extremely rapid, suggesting that the first minimum is likely to be both pronounced and unique in practical cases.

We are now ready to define the OFSS. It is necessary to make the definition relative to $\theta_k$, as the optimal value varies with $k$. We discuss the issue of the choice of $k$ later.

Definition: OFSS$(k, A)$ is FSS$(\eta^*; A)$, where $\eta^*(k; \alpha) \in (0, 1]$ is the value that minimizes $I_k(\alpha')$, subject to $\alpha' \geq \alpha'_{k*}$.

Even if there are rare cases when $\alpha^* \neq \alpha^{**}$, there are advantages to basing a definition on the first minimum. These stem essentially from the following property.

$\textbf{Theorem 9:}$ The function $p_f^*(k; \alpha)$ takes the form

$$p_f^*(k; \alpha) = \begin{cases} 1, & \alpha < \alpha'_{k*} \\ \frac{\alpha}{\alpha'}, & \alpha \geq \alpha'_{k*}. \end{cases} \tag{28}$$

Proof: If $\alpha \geq \alpha^*$, then the available range $[0, \alpha]$ for $\alpha'$ includes $\alpha^*$, which can be attained by setting $\eta^* = \alpha^*/\alpha$. Since $\alpha^*$ is the first local minimum of $I_k(x)$ over $x \geq 0$ and $I_k(x)$ is continuous, this gives the minimum for this $\alpha$, namely $\mathcal{I}(\eta^*) = I_k(\alpha^*)/A$, under the constraint $\eta^* \leq \alpha'_{k*}$. If $\alpha < \alpha^*$ then $\alpha^* \not\in [0, \alpha]$ so $\alpha^*$ cannot be achieved. Since $\alpha^*$ is the first local minimum, minimal variance over $[0, \alpha]$ lies at the endpoint $x = \alpha$ which is achieved at $\eta^* = 1$.

Equation (28) tells us that for fixed $\alpha$ and any given $k$, the form of $p_f^*$ as a function of $\alpha$ is not only simple and explicitly known, it is determined by a single real parameter $\alpha'_{k*}$, and is independent of the number of counters $A$.

If $\alpha^* < \alpha^{**}$ and OFSS were based on $\alpha^{**}$ instead of $\alpha^*$, then $p_f^*(\alpha)$ would no longer be a non-increasing function, determined by $\alpha^{**}$ alone, but instead a function with discontinuities at each minima up to and including $\alpha^{**}$.

B. Exploring $\alpha^*$

In this section we explore how $\alpha^*$ and $p_f^*$ behave as a function of $\alpha$ and $\theta$ for each index $k \in [W]$. The evaluation of $\alpha'_{k*}$ is described in Section V.D.

Our main focus is on distributions $\theta$ with non-increasing densities, exemplified through geometric distributions truncated at $k = W$ (and renormalized). We control the decay rate $\lambda$ through the tail decay ratio $R = \eta(1)/\eta(W)$ (corresponding to $\lambda = \log(R)/(W - 1)$), which enables direct specification of the degree of tail drop over the support. This class, which we denote $TG(W, R)$, includes both very...
Fig. 1. Illustrating \( p_\alpha^*(k) \) as a function of \( \alpha \). Left: \( k = 1 \) for uniform \( \theta \) with \( W = 10 \) (\( \alpha_\alpha^* \approx 1.0034 \)). Center: \( k = \{1, 2, 10, 100\} \) for uniform \( \theta \) with \( W = 100 \), (\( \alpha_\alpha^* \approx \{0.9999, 0.9946, 0.9552, 0.6983\} \)). Right: \( k = 1 \) for uniform and truncated geometric \( \theta \), each with \( W = \{100, 1000\} \) (respectively \( \alpha_\alpha^* \approx \{0.9999, 1.0000\} \) for Uniform, \( \alpha_\alpha^* \approx \{0.9765, 0.9777\} \) for TG(\( W, 1000 \)).

Fig. 2. Exploring \( \alpha^*(k) \) as a function of \( k \). Left: Uniform and TG(\( W, 1000 \)) for \( W = \{10, 20\} \). Middle: the same for \( W = \{100, 200\} \), and Right: for \( W = \{1000, 2000\} \). We see that the critical value is monotonic in each of \( k \), the distribution support \( W \), and the tail decay \( R \) parameters.

heavy tails when \( R \approx 1 \) (\( R = 1 \) is the uniform distribution) as well as geometrically decaying tails with negligible mass at \( k = W \) when \( R \) is large. For our purposes it is a reasonable model class for the flow size distribution. The next three figures are based on the TG(\( W, R \)) class.

Figure 1 provides some examples of \( p_\alpha^*(\alpha) \). The left plot is a simple reference example, a uniform \( \theta \) with \( W = 10 \), looking at the optimum sampling rate for \( k = 1 \) where the density is largest. We find that \( p_\alpha^* \) must be set below \( p_\alpha^* = 1 \) when \( \alpha \) exceeds \( \alpha_\alpha^* \approx 1.0034 \) in this case, and the graph is generated from (28) using this value. In the middle plot we increase the support to \( W = 100 \), and observe how \( p_\alpha^*(k; \alpha) \) varies with \( k \). The main point to note is that each \( k \)-curve maintains a fixed relationship with all others for all \( \alpha > \alpha_k^* \).

In this case we see that curves with larger \( k \) lie lower, corresponding to smaller \( \alpha^* \). The final plot returns to \( k = 1 \) and compares results for two very different distributions: uniform, and TG(\( W, R = 1000 \)), each for \( W = 100 \) (gray scale curves) and the more realistic \( W = 1000 \) (colored curves). The four curves are remarkably close, reflecting in each case \( \alpha_k^* \) values close to 1.

Figure 2 examines the dependence of \( \alpha_k^* \) on \( k \) in more detail. In each plot results are given for two uniform distributions and two TG distributions with \( R = 1000 \), each for two similar values of \( W \). In the left plot \( W = 10, 20 \) are used, and all \( k \) values are plotted. The middle plot shows \( W = 100, 200 \) and 24 values over \( k \in [1, 200] \) are plotted, and in the right plot \( W = 1000, 2000 \) and 15 values are plotted over \( k \in [1, 2000] \). These samplings of \( k \) space are sufficient to show the shape of \( \alpha^*(k) \) adequately. In all cases we see that \( \alpha_\alpha^* \) is the highest value and is very close to 1.

Figure 3 examines the dependence of \( \alpha^* \) on \( \theta \) more directly by replotting prior \( \alpha_k^* \) results as a function of \( W \), supplemented with new \( (W, R) \) pairs with larger \( W \) and \( R \) values.

In summary, for the monotonic \( \theta \) family we observe monotonic behavior of \( \alpha^* \) in several parameters, including monotonic decline with increasing \( k \) provided \( W \) is sufficiently large, and also with increasing \( R \) for fixed \( W \) and \( k \). This is not unexpected since making any of \( k \), \( R \) or \( W \) larger whilst fixing the others results in a lower \( \theta_k \).

A smaller parameter value is nominally harder to estimate, and if so its estimation should be more sensitive to the level of ‘ambiguity’ in the sketch, resulting in better performance at lower load levels, and hence to a smaller \( \alpha^* \). The rise at \( k = W \) observed for the Uniform examples on the left of Figure 2 can be explained by integer effects: because \( k > W \) is impossible, it is relatively more likely at \( k = W \) that events such as \{counter value \( C = mW \)\} arise through \( m \) flows of size \( W \), a simple combinatoric from which \( \theta_k \) can be well estimated.

In general, the value of \( \alpha_k^* \) is a complex function depending on the detailed discrete properties of \( \theta \). Figure 4 illustrates how non-monotonic \( \theta \) can result in decidedly non-monotonic \( k \) dependence. Figure 5 illustrates that \( \alpha^* \)
values well over 1 are not hard to find, and that the largest \( \alpha_k^* \) is not necessarily at \( k = 1 \).

C. Efficiency

In this section we examine the CRLB performance of OFSS, and examine its relative efficiency compared to that of its FS and Sk components. Since the number of counters enters in only as a multiplicative factor, we plot per-counter variance rather than total variance, for all figures and quantities in this section.

Figure 6 provides an example of the per-counter constrained CRLB for OFSS, and its underlying components FS and Sk, as a function of \( \alpha \). Each curve is a variance bound for \( \theta_1 \) for a uniform \( \theta \) with \( W = 100 \).

The top (black) curve is of \( I_{FS}^- \), where the minima at \( \alpha_1^- \approx 1 \) is clearly seen. By construction, OFSS is identical to Sk for \( \alpha \) up to \( \alpha_1^- \), and is held constant thereafter (blue curve). The lowest possible variance is obtained when all flows are sampled perfectly, this is equivalent to FS with \( \rho_f = 1 \) and is given by \( \theta_1(1 - \theta_1)/\alpha \), the thick gray curve. Finally, the red dashed curve shows \( FS(p_f^*(1, \alpha)) \), which represents a lower bound to the variance of OFSS by ignoring its sketch component.

There are two performance measures of interest, on display in Figure 6 which are ’internal’ to OFSS. The first is an evaluation of the extent to which OFSS succeeds in its aim to be an FS-like method which is implementable in practice, and the second is the degree of sensitivity to a miscalibration of \( \alpha^* \).

We examine the relative efficiency of OFSS compared to its FS first stage in the ‘fully loaded’ regime where \( \alpha \geq \alpha^* \). That is, we calculate \( \frac{(I_{FS}^-(1))_{kk}}{(I_{FS}^-(\alpha^*))_{kk}} = \frac{\theta_k(1 - \theta_k)}{(I_{Sk}^-(\alpha^*))_{kk}} \) corresponding to the ratio of the values of the red and blue horizontal lines from Figure 6.

\[ \frac{CRLB(\theta_1)}{FS(1)} \]

\[ \frac{CRLB(\theta_1)}{Sk = FSS(1)} \]

\[ OFSS \]

\[ FS \text{ stage of OFSS} \]
Results are given for a number of TG(W, R) models in the left plot in Figure 7. The results are remarkably independent both of model and index k. In each case, OFSS has an efficiency of around 1/3 compared to what is initially collected by its FS filtering stage. This is not surprising since, by definition, the effect of collisions within the sketch component are far from negligible at the optimal loading. This makes intuitive sense when α∗ = 1, since then around a third of counters are empty, the corresponding flows causing collisions in other counters. Since most colliding counters will only have 2 flows in them, this leads to around a third of counters with exactly one flow, and these are the ones carrying the most valuable information on θ. In the estimation context, the importance of accounting for collisions (and in a computationally effective manner) was recognized in the ‘fast estimator’ from [15, pp. 77–81].

The above loss of efficiency is tolerable, and its insensitivity to parameters offers a simple way to approximately calibrate methods based on a desired variance. Furthermore, since FS has the highest performance among methods studied to date [20], this suggests that OFSS performs to within a constant factor of the best known method, yet is implementable even at the highest speeds within core Internet routers.

From Equation (23) and Section III-D we know that the relative efficiency of an uncalibrated FSS, in particular Sk=FSS(1), compared to OFSS will tend to zero with increasing α. The right plot in Figure 6 explores the impact on efficiency of a more moderate miscalibration by comparing OFSS to an ‘OFSS’ miscalibrated using 2α∗ instead of α∗. That is, we calculate \((\mathcal{I}_{FSS(\alpha^*)})_{kk}/(\mathcal{I}_{FSS(2\alpha^*)})_{kk}\). Using the same θ models as in the left plot, we again find a remarkable insensitivity to both the model and index k. In each case, the miscalibration resulted in an efficiency of around 3/4 compared to OFSS. This is good news for practitioners, since the data (or models) used to calibrate the method will typically not be those of the data stream being measured, and so the α∗ value(s) being used will never be exactly right. It also means that the method can be used on unknown data using nominal values for α∗, such as α∗ = 1, without fear of a catastrophic drop in performance. This will be sufficient for the purposes of an initial estimation of θ, from which more accurate α∗ can subsequently be obtained.

Very similar results were found for the non-monotonic examples of Figure 6 (sketch efficiency over all k clustered about the mean 0.38, and about 0.71 for mis-calibration), and Figure 5 (0.35 and 0.77 respectively).

D. Evaluation of \(p_j^k\)

As described above, there is no single optimum value of the flow sampling parameter \(p_j\) for a chosen k and given α. By (28), this reduces to the evaluation of \(α_k^∗\). This avoids the need to numerically evaluate the function at each α value separately, in particular at large α values where computation times and numerical errors increase dramatically. This is also an advantage operationally as α is the obvious external variable which may change from one measurement interval to the next. By knowing \(α_k^∗\) for a chosen k of interest, the \(p_j^k(α)\) can track α without the need for significant extra computation.

We begin by describing the evaluation approach, and then consider the important practical and methodological issue of W truncation.

1) Calculating \(α_j^k\): We evaluate \(α_j^k\) by numerically determining the first minimum of \(I_k(x) = (\mathcal{I}_{Sk}(x)/x)_{kk}\). Thus it is necessary to evaluate \(I_k(x)\) at many different values of x.

The first step is to evaluate the counter density \(c_j(θ, x)\). For this we use successive convolution based on (9). Let \(f_k\) denote a Poisson density with parameter \(λ_k = αθ_k\). We initialize with \(c = f_1\), and then recursively calculate \(c = c \otimes (\uparrow_k f_k)\) over \(k = 2, 3, \ldots, W\), where \(\uparrow_k\) is an upsampling operator which places \(k − 1\) zeroes between each element of its argument. To limit vector length, at each step we truncate the tail of c to remove elements whose values are at machine precision limits or below. We investigated the much faster alternative of calculating the
above convolutions using the Fast Fourier Transform, and also by calculating the generating function inverse of \( I(k) \) directly. In either case however numerical artifacts were observed in the distribution tail.

The next step is to evaluate \( J_{sk}(x) \) which depends on the density. Our calculation is based on a truncation of the matrix sum (15) (ignoring the \( I_W \) term, which is subsequently nullled). The truncation occurs adaptively when the largest entry in the current increment-matrix falls below a tolerance \( Itol \) which we set to \( 10^{-7} \). We have observed a close correspondence between \( Itol \) and the absolute error of the final value of \( I_k(x) \). This step is the bottleneck of the calculation and has time complexity of \( O(W^2) \).

The next step would seem to be to evaluate \( T_{sk}^+(x) \) from \( J_{sk}(x) \) using (3), which would give access to \( I_k(x) \) for all \( k \). However, as this involves calculating \( J_{sk}^{-1}(x) \), it has high complexity. More importantly, the sequence of \( x \) values needed in the minimum finding differs for each \( k \), so it is not of immediate value to evaluate all \( k \) for a given fixed \( x \). Consequently, we directly calculate \( I_k(x) \) for the chosen \( k \) only. For this we can use the Gauss-Seidel procedure which is more time and space efficient, and provides better numerical stability compared to full matrix inversion. Let \( e_k \) be the kronecker vector for index \( k \). Performing Gauss-Seidel on the equations:

\[
J_{sk}(x)x = I_W, \\
J_{sk}(x)y = e_k,
\]

yields \( x = J_{sk}^{-1}(x)I_W \) and \( y = J_{sk}^{-1}(x)e_k \) respectively. The \( k \)-th diagonal term of (3) then reduces to

\[
e_k^T J_{sk}(x)e_k = e_k^T x - e_k^T x x^T e_k / (1 - e_k^T x)
\]

and dividing by \( x \) we obtain the desired \( I_k(x) \).

Clearly, if \( I_k(x) \) were calculated for all \( k \in [1, W] \) then the time complexity would be multiplied by \( W \). In practice however only a few \( k \) are likely to be sufficient, in particular for specific pre-determined tasks such as the tracking of network anomalies or attacks of known flow-size signature. The values \( k = 1, 2 \) are of particular importance.

To locate the desired minima, \( x = \alpha_k^* \) of \( I_k(x) \), we use the golden section search in Matlab’s \texttt{fminbnd} routine to search in the interval \( x \in [0, \alpha] \), using a convergence criterion of \( TolX = 10^{-6} \). We initially search with \( \alpha = 1.1 \) as we observed that typically \( \alpha^* < 1 \) for the distributions we evaluate, and it is desirable to keep \( \alpha \) as small as possible to reduce both computation time and numerical instability. If we find \( \alpha^* = \alpha \) (to within \( 2TolX \), we set \( \alpha \rightarrow 2\alpha \) and repeat the search with a wider search window, and continue this procedure until a non-trivial \( \alpha^* < \alpha \) is found. For \( \alpha > \alpha^* \) convergence was consistently reached in approximately 10 iterations at \( k = 1 \), down to 7 iterations for \( k = W \). The convergence time is not sensitive to \( \alpha \) provided it is not very close to or below \( \alpha^* \).

2) \( W \) Truncation: In applications such as flow sizes in data networks the maximum flow size \( W \) can be large, for example well over a million. The calculation of the CRLB, and \( \alpha_k^* \) in particular, can be prohibitive under these circumstances. There are also modeling aspects to consider. In real data there is no natural choice of maximum flow size, and file sizes are increasing over time. Moreover, selecting \( W \) to be an outlier of the flow size distribution means that the condition \( \theta_k > 0 \) will be violated for many \( k < W \). These considerations point to the need to truncate \( W \) to reasonable values in practice.

In this section we investigate the impact of truncation on the evaluation of \( \alpha_k^* \). More precisely, we define a truncated distribution \( \theta_W \) with support \([W]\) as follows:

\[
\theta_k = \frac{\theta_k}{\sum_{k=1}^{W} \theta_k}, \quad k \in [W], \text{ else } \theta_k^{W} = 0.
\]

We define the relative absolute error in \( \alpha_k^* \) as

\[
\text{rel-err}_k = \frac{|\alpha_k^* - \alpha_k^{W}|}{\alpha_k^*},
\]

where \( \alpha_k^* \) is the critical value of OFSS for \( \theta_W \).

In Figure 8 we evaluate rel-err for some examples in the TG family. In the top plot \( W = 100 \), and we consider a number of truncations from \( W = 10 \) to \( W = 95 \), for each of \( R = \{1, 1000, 1000000\} \). For each TG(\([W], R\)) a curve is plotted showing the error for a number of \( k \) values in the range \([W]\). The important observation is that the critical values for \( k \ll W \) are not strongly affected by truncation. The indices for which the error is acceptable expand rapidly for higher \( R \) (faster tail decay), as one might expect. The bottom plot shows a more realistic cases with \( W = 1000 \). For example when truncating TG(1000, 1000) down to an easily manageable TG(\([100], 1000\)), the relative error at \( k = 1 \) is only 0.015%, and even the worst errors (at \( k = W \)), are correct to within a few percent for all \([W]\). For the more rapidly decaying \( R = 1000000 \) distribution errors are negligible in most cases.

We conclude that for roughly monotonic distributions with reasonable tail decay, such as in computer network flow sizes, it is justifiable to calculate \( \alpha_k^* \) on truncated distributions for the \( k \) of most interest, in particular \( k = 1 \).

Code calculating \( \alpha_k^* \) for the TG class used here or arbitrary \( \theta \), and the corresponding Fisher Information and CRLB matrices, is available at [23].

VI. THE EVICTION SKETCH ES\( K \)

In [15], [13] Ribeiro et al introduced a data collection method for the flow size distribution, which is in fact a skampling method, which we will refer to as the Eviction Sketch (ESk). In this section we describe ESk and provide the first published Fisher Information based analysis of it.

A. Definition

We can think of ESk as an Sk which has been physically enhanced by associating to each of its \( A \) counters an
Fig. 8. Absolute relative errors in $\alpha^*$ using truncated TG models with tail decay ratios $R = \{1, 1000, 1000000\}$ for Top: $W = 100$ with truncations $W \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 95\}$; Bottom: $W = 1000$ with truncations $W \in \{100, 200, 400, 600, 800\}$. For each truncated TG($W, R$) a curve shows the error for a subset of remaining indices. We see that errors are small unless the distribution $\theta$ is uniform ($R = 1$) and the truncation $W$ is low.

**Ownership variable** taking values in $[L]$, all initialized to $L$. Incoming flows are given a random priority class $\ell \in [L]$. If $\ell$ equals the value of the associated ownership variable its counter is incremented as normal when a packet from the flow arrives, but if it is larger (lower priority than it) the counter is unchanged, and if smaller (higher priority) the counter is reset to 1, and the ownership variable is set to $\ell$ recording the fact that class $\ell$ now ‘owns’ the counter. This random eviction of lower class packets is an implicit sampling, hence ESK is a skampling method.

Although the values of the ownership variables depend on the order of flow arrivals during the measurement interval, their final states at the end of the interval do not. This is critical as it means that, just as for FSS and FS, the method can be analyzed without requiring knowledge of packet arrival order. Moreover, as for FSS and FS, if a flow is sampled at all, all its packets will be, so it remains meaningful to say that the sampling reduces the input flow load $\alpha$ to an effective load $\alpha' < \alpha$.

**B. Fisher Information**

Let $C^{(\ell)}$ denote the number of packets of class $\ell$ mapping to a given fixed counter, prior to any eviction. It is easy to see that such ‘virtual counters’ are i.i.d., each distributed as a counter from $S_k$ with load $\bar{\alpha} = \alpha / L$. Define $\bar{\lambda}_k = \bar{\alpha}\theta_k$, $\bar{\Lambda} = \sum_{k=1}^{W} \bar{\lambda}_k = \bar{\alpha} \sum_k \theta_k = \Lambda / L$, and set $E = e^{-\bar{\Lambda}}$. Analogously to before $\bar{\Lambda}$ reduces to $\bar{\alpha}$ under the constraint. In terms of the densities themselves we may set $\bar{\Lambda} = \bar{\alpha}$ immediately, however as before the unconstrained forms are needed for the purpose of calculating the unconstrained Fisher Information $J$.

The joint density $w_{j, \ell}$ of $(C, O)$ is as follows. For $j > 0$

$$w_{j, \ell} = \Pr\left(\{C^{(i)} = 0, \forall i < \ell\}, C^{(\ell)} = j\right)$$

$$= \left(\prod_{i=1}^{\ell-1} \Pr(C^{(i)} = 0)\right) \Pr(C^{(\ell)} = j)$$

$$= E^{\ell-1} c_{S_k, j}(\bar{\alpha})$$

where $c_{S_k, j}(\bar{\alpha})$ is the density (12) of $S_k$ with load $\bar{\alpha}$. When $j = 0$ no packets have entered the sketch, and so $O$ remains at its initial value of $O = L$. Hence $w_{0, \ell} = 0$ for $\ell < L$, and
Fisher matrix for the joint observable is given by

\[ w_{0,L} = E^L = e^{-\Lambda}. \]

It is easy to verify that the marginal density of \( O \) is

\[ w_{\ell} = E^{\ell-1}(1-E), \quad \ell \leq L, \quad (29) \]

\[ w_{L} = E^{L-1}, \quad (30) \]

and that of the counter density \( C \) is

\[ c_0 = e^{-\Lambda}, \quad (31) \]

\[ c_j = F(L) c_{Sk,j}(\alpha/L), \quad j \geq 0, \quad (32) \]

where \( F(L) = \sum_{\ell=1}^{L} E^{\ell-1} = (1 - e^{-\Lambda})/(1 - e^{-\Lambda/L}) \) is a normalization factor obeying \( F(L) \geq 1 \) which is monotonically increasing in \( L \).

Despite its relatively complex eviction-based construction, the resulting density \( c_j \) can be interpreted as a simple mixture distribution combining an atomic mass at \( j = 0 \) selected with probability \( e^{-\alpha} \), with a counter variable conditioned on \( j > 0 \) (with density \( c_{Sk,j}(\alpha)/(1-e^{-\alpha}) \)) selected with probability \( 1 - e^{-\alpha} \).

Clearly \( \mathbb{E}[C] = F(L) \mathbb{E}[C_{Sk,j}(\alpha)] = F(L)\alpha D \). Consequently, the average per-packet sampling rate of \( ES_k \) is \( p = \mathbb{E}[C]/(\alpha D) = F(L)/L \), and so \( \alpha' = p \alpha = \alpha F(L)/L \). For \( L = 1 \), ES\( k \) reduces to \( Sk \), a pure sketch with \( p = 1 \).

We can show (see Appendix C) that the per-counter Fisher matrix for the joint observable is given by

\[ (J)_{sk} = \frac{\alpha^2}{L^2} H + F(L) \alpha \left( J_{Sk} \left( \frac{\alpha}{L} \right) \right)_{sk} \quad (33) \]

where \( H = H(\alpha, L) \) is independent of \( i, k \). In matrix form, this constant give rise to a term \( H 1_W 1_W^T \) which is subsequently nullled by the constraint in the formation of the constrained CRLB, and so can be ignored (the nulling follows from \( 4 \)) and the fact that \( U^T 1_W = 0 \), since the columns of \( U \) span the null-space of \( 1_W \).

The per-flow information is of course just

\[ J_{ESk}(L, \alpha) = AJ/N_f = J/\alpha, \quad (34) \]

and from \( 33 \), the constrained per-flow CRLB is given by

\[ \mathcal{I}^{+}_{ESk}(L, \alpha) = \frac{L}{F(L)} \mathcal{I}^{+}_{Sk} \left( \frac{\alpha}{L} \right) = \frac{1}{p} \mathcal{I}^{+}_{Sk} \left( \frac{\alpha}{L} \right). \quad (35) \]

Finally, the per-counter CRLB is

\[ \mathcal{I}^{+}(L, \alpha) = \frac{\mathcal{I}^{+}_{ESk}(\alpha/\alpha)}{\alpha} = \frac{1}{\alpha'} \mathcal{I}^{+}_{Sk} \left( \frac{\alpha}{L} \right). \quad (36) \]

Since the class of a packet is determined uniformly at random, it is clear that \( O \) carries no information about \( \theta \). The value \( j \) of the counter variable \( C \) is then a sufficient statistic for \( \theta \) (this is immediately verified by observing that the joint density obeys the Fisher-Neyman factorization, or alternatively that \( \text{Pr}(O = \ell|C = j) \) is independent of \( \theta \)). Thus, although the ownership variables are needed to implement ES\( k \), they do not constitute a source of additional information about the flow size, once the constraint is applied.

C. The \( \alpha \) Dependence of \( ES_k \)

In ES\( k \) the degree of sampling \( p \) is implicit rather than set by an explicit parameter. It is also harder to control as it varies with \( \alpha \), unlike \( p_f \) which is fixed for FSS, and the control parameter, \( L \), only takes integer values. For fixed \( L \) it decreases monotonically with \( \alpha \), from \( p = 1 \) at \( \alpha = 0 \) down to a limit of \( 1/L \) for \( \alpha \gg L \). Thus although \( p \) adapts in the right direction as load increases, it is too small for small \( \alpha \) (\( p = 1 \) would be better) and too large for large \( \alpha \) (need \( \alpha' \approx \text{constant} \) rather than \( \alpha' \approx \alpha/L \)). Thus ES\( k(L) \) for any \( L \) has the same fundamental limitation as FSS\((p_f)\), beyond a certain load range it will be overloaded.

It is easy to see that \( \alpha p = \alpha' > \tilde{\alpha} = \alpha/L \), since \( \alpha' \) will pick up the load \( \tilde{\alpha} \) from the highest priority virtual counter, as well as load contributions from lower priority counters when they end up owning the physical counter. However for \( \alpha \gg L \), with \( L \) held fixed, \( \alpha' \) tends to \( \tilde{\alpha} \) from above since then each physical counter will contain packets from, and hence be owned by, the highest priority flows. From \( 36 \), the CRLB and the physical sketch of ES\( k \) will then look exactly like that of \( Sk(\tilde{\alpha}) \) with \( \tilde{\alpha} \approx \alpha' \gg 1 \), a reduction in overload by \( 1/L \), but overloaded nonetheless.

The proportion of empty counters is given by \( e^{-\alpha}, \) exactly as for \( Sk \).

VII. THE OPTIMIZED EVICTION SKETCH OESK

In this section we investigate the dependence of the per-counter CRLB \( 35 \) on the internal parameter \( L \). In \( 16 \) it was stated that \( L = 2 \) was optimal. Here we show that much larger values give much better, in fact FS-like, performance. Effectively, we define a new optimized method based on ES\( k \), called OES\( k \), much as we defined OFSS based on FSS.

A. The Nature of ES\( k \) Sampling

The nature of the flow thinning in ES\( k \) is very different to that of FSS. Rather than sampling then sketching (which treats all flows equally), ES\( k \) first sketches, then samples in a conditional sense. As the sampling is based on priority only, not on the number of flows, for non-empty counters the ‘clearing out’ of collisions therefore increases monotonically with \( L \), but has the important deterministic property that it will never empty a counter completely. In fact for large enough \( L \) there will only be a single flow in any given class, and hence exactly one flow per non-empty physical counter. This is reflected in the expression for \( p \), which for fixed \( \alpha \) and \( L \gg \alpha \) obeys \( p \approx (1 - e^{-\alpha})/\alpha \), which corresponds precisely to reducing the load in all non-empty counters to exactly 1. In other words, ES\( k(\infty) \) is equivalent to Flow Sampling with \( A(1 - e^{-\alpha}) \) counters.

For simplicity, we now focus on the important regime of large \( \alpha \), but maintaining \( L \gg \alpha \). In this case there are effectively no empty counters, and ES\( k(\infty) \) is equivalent to FS\((1/\alpha)\). In this regime \( \alpha' \to 1 \) and \( \tilde{\alpha} \to 0 \), and so the per-counter variance \( 36 \) is just \( \mathcal{I}^{+}(\infty, 0) = \mathcal{I}_{Sk}(0) = \theta_k(1 - \theta_k) \), just as for FS.
Since increasing $L$ clears collisions monotonically without changing the number of empty counters, we expect that for any reasonable $\theta$ the variance decrease will likewise be monotonic, tending in the limit to FS which we know from prior work [20] is (pathological cases aside) optimal.

We have just shown that ES$k(L)$ is optimal at $L = \infty$ where it reduces to FS with a flow-table of size $A$. However, we have thus far ignored the additional memory use inherent in the ownership variables, and $L = \infty$ implies infinite additional memory! The next section re-examines the question of optimal $L$ taking this into account.

**B. Defining an Optimal $L^*$ and OES$k$**

We will define an optimal $L = L^*$ by adding a constraint, $M$, the total amount of memory (in bits) used by ES$k$. This creates a tradeoff whereby larger $L$ implies smaller $A$. Since $A \geq 1$, we have $L \leq \lfloor 2M - b \rfloor$. It is necessary to specify the number of bits $b$ used by each counter. Here and below we assume that $b$ is fixed, and is large enough so that counter overflow can be ignored.

Since each ownership variable requires $\log_2 L$ bits, we have $M = A(b + \log_2 L)$ (we allow $A$, $L$ and $\log_2 L$ to take continuous values for simplicity). Hence $A = M/(b + \log_2 L)$ is now determined by the independent variable $L$, and so the per-counter load $\alpha(L) = N_1/A(L)$ is an increasing function of $L$. Note however that $\tilde{\alpha} = \alpha/L = \alpha(1)(1 + (\log_2 L)/b)/L$, still tends to zero with $L$, albeit at a slower rate.

**Definition: OES$k(k, M)$** is ES$k(L^*_k; M)$, where $L^*_k(\alpha) \geq 1$ is the value that minimizes $(\mathcal{I}^+(L)/A(L))_{kk}$ at constant $\alpha$, subject to total memory $M$.

Because the tradeoff is based on varying the number of counters, we cannot use per-counter variance as a surrogate for total variance as we did before. It follows that the notion of an optimal $L^*$ is more complex than that of $p_f^*$, in that the optimum cannot be expressed in terms of a scaled load $\alpha'$ or even $\tilde{\alpha}$ which is dependent only on $\theta$. Instead $L^*$ is a function of $\alpha(1) = bN_1/M$ and $b$ as well as $\theta$.

We derive an approximate expression for $L^*$ in the regime of large $\alpha$ and $L \gg \alpha$. Here can assume that a counter has either a single flow, with probability $p_1$, or two flows, and so write $p_\alpha = p_1 + 2(1 - p_1)$, whence $p_1 = 2 - p_\alpha$. A relevant expansion for $p$ is $p = \frac{1}{2}(1 + \tilde{\alpha} + O(\tilde{\alpha}^2))$, and so $p_1 \approx 1 - \tilde{\alpha}/2$. The idea is simply to keep track of the number of counters with a single flow, because others would not typically be substantially used by an optimal estimator. There are approximately $Ap_1$ of these with a corresponding total variance

$$\frac{\theta_k(1 - \theta_k)}{Ap_1} = \frac{(1 + (\log_2 L)/b)\theta_k(1 - \theta_k)}{1 - \tilde{\alpha}/2} \frac{M/b}{A(1)/A(L)} \frac{A(L)}{A(1)} = (1 + (\log_2 L)/b) \frac{\theta_k(1 - \theta_k)}{1 - \tilde{\alpha}/2} \frac{M/b}{A(1)/A(L)} \frac{A(L)}{A(1)} = (1 + (\log_2 L)/b) \frac{\theta_k(1 - \theta_k)}{1 - \tilde{\alpha}/2} \frac{M/b}{A(1)/A(L)} \frac{A(L)}{A(1)}$$

(37)

The tradeoff is clearly seen in the factor $K(L; \alpha(1), b) = (1 + (\log_2 L)/b)$. As $L$ increases the denominator increases rapidly toward its limit (collisions all but cleaned out) resulting in decreasing then saturating variance, at which point the numerator, with its slower $\log_2 L$ growth (smaller $A$ as the ownership values take more bits) takes over causing variance to diverge in the limit. We can approximate $L^*$ as the unique minimum of this function, $\tilde{L}^*$, which we evaluate numerically.

Since the above argument is at the level of granularity of flows and does not involve $\theta$, the $\tilde{L}^*$ is likewise independent of $k$. We expect it to be most relevant for the uniform distribution, where the $k$ dependence is weakest, and where the assumption of ignoring counters with more than one flow holds best. This is borne out in Table III which provides some examples of $L^*_1$ and its closeness to $\tilde{L}^*_1$. In the uniform examples the agreement is extremely close. For non-uniform cases the agreement is poor at small $W$ but excellent in the practically important large $W$. In terms of the error in the predicted number of bits needed for the ownership variables (third row) however, the error is in all cases perfectly adequate for the purpose of understanding the tradeoff. Table III gives the results for $\theta = 1$. As expected, estimation is more difficult in the distribution tail, the more so for higher tail decay ratio $R$, and so requires a higher $L$ corresponding to even fewer collisions.

**C. Efficiency**

The term $\frac{\theta_k(1 - \theta_k)}{Ap_1}$ from Equation (38) is just the variance of flow sampling with exactly $A(1) = M/b$ flows. Hence the efficiency of OES$k$ with respect to FS can be approximated by

$$\tilde{\text{Eff}} = \frac{1}{K(L^*)} = \left(1 - \tilde{\alpha}^*/2\right) \frac{A(L^*)}{A(1)} .$$

(39)

This expression factors the efficiency $\tilde{\text{Eff}} < 1$ into a drop $(1 - \tilde{\alpha}^*/2) < 1$ due to a small number of counters containing more than one single flow, and a drop $A(L^*)/A(1) < 1$ due to the reduced number of counters induced by the memory constraint. The last two rows of Tables II and III provide $\tilde{\alpha}^*$ as well as the true efficiency $\text{Eff} < 1$. These are consistent with the intuition underlying $L^*$ that most of the efficiency drop is due to counter loss. They also confirm the expectation, since $A(L)$ decreases slowly as a function of $\log_2 L$, that ES$k$’s collision clearing can be highly effective even for moderate $L$, and hence that the efficiency of OES$k$ is in general high.

**VIII. COMPARING OFSS AND OES$k$**

We begin with a result comparing FSS with ES$k$ on the basis of equal counter memory $A$, and matched sampling rate, which is independent of $b$ and fair in the sense of the number of captured packets (the ‘ESR’ normalization used extensively in [20] in comparisons of sampling methods). Since each method is closely related to Sk, a strong positive semidefinite result is possible.

**Theorem 10:** For any $L$ and for any $\alpha > 0$, $\mathcal{I}^+_{\text{FSS}} > \mathcal{I}^+_{\text{ESk}}$ when $p_f = p = F(L)/L$, and the same is true for total variance.
Proof: By selecting $p_f = p = F(L)/L$, the CRLB expressions \((25)\) for $\mathcal{I}_{\text{FSS}}$ and \((35)\) for $\mathcal{I}_{\text{ESk}}$ differ only in the argument, respectively $\alpha' = p\alpha$ and $\bar{\alpha} = \alpha/L$, of $\mathcal{I}_{\text{ESk}}$. It is simple to show that $p - 1/L = (e^{-\alpha} - e^{-\bar{\alpha}})/(L(1 - E)) \geq 0$. Hence $\alpha' \geq \bar{\alpha}$ and the per-flow result follows immediately from the monotonicity of per-flow CRLB (Theorem \(3\)). Since $\alpha$ and $A$ are the same for each, the inequality also holds for per-counter and total variance.

Since $\alpha' \to \bar{\alpha}$ in the limit of large $\alpha$, it follows that $\text{FSS}(p(L))$ and $\text{ESk}(L)$ have to first order the same (terrible) variance in the limit.

Theorem \(10\) shows that ESk is in a sense always more efficient than FSS, but it does not imply the same between OESk and OFSS, for two reasons. First, because FSS was forced to adopt a sampling probability $p_f = p$ driven by $L$, which may be far from $p_f^*$, and second, because here ESk was given an unfair memory advantage. To compare the two optimized methods directly, we need a new approach.

A. Efficiency

In this section we follow Section \(\text{VII}\) and compare on the basis of total shared memory $M$.

We compare the methods by comparing their efficiencies relative to FS, described in Section \(\text{V-C}\) for OFSS, and Section \(\text{VII-C}\) for OESk. For OESk, efficiency is mainly controlled by the reduced number $A(L^*)$ of counters, since only a small proportion (of the order of $\bar{\alpha}^*/2$) of available counters are (approximately) unusable. In contrast, OFSS reduces collisions by thinning at ingress, which reduces collisions, but at the cost of increasing the proportion $e^{-\alpha^*}$ of counters which are empty and therefore wasted, while still having a non-negligible proportion which have two or more and therefore unusable.

The comparison of total variance therefore reduces to comparing the number of counters which carry a single flow: namely $A(1)c_1(p_f^*\alpha(1))$ in OFSS, and $A(L^*)$ in OESk. Consider first $\theta_1$. For TG densities an efficiency of around 1/3 was reported above for OFSS, and from Table \(\text{II}\), values around twice this for OESk. However, since the comparison depends on a number of parameters, including a very strong dependence on $b$ as well as $\alpha(1) = bN_f/M$, no universal conclusion can be drawn. For example it is not difficult to see that the efficiency of OESk drops to zero as $\alpha$ increases, since OESk can only reduce its sampling rate to $p(L) \approx 1/\alpha$ to match $\alpha$ by allocating more memory to increase $L$. For example a value of $\alpha(1) = 1000$ with $b = 16$ causes efficiency to drop to around 0.5. In contrast, the memory consumption of OFSS, and its efficiency, is a constant regardless of its sampling rate $p_f$.

Now consider $\theta_W$. Recall from Figure \(7\) that the efficiency of OFSS dropped monotonically with $k$ and was smallest for $k = W$, but that the drop was small, in particular for the uniform cases. Table \(\text{III}\) suggests a very similar picture for OESk: very similar efficiencies are obtained as for $\theta_1$, however more sampling (higher $L$) is required to achieve them. The accuracy of the approximation $\hat{L}^*$ to $L^*$ drops substantially for $k = W$, however in terms of bits of storage it is still sufficient to give the correct order of magnitude.

In conclusion, from an efficiency standpoint, each method is very strong, which is better being a function of $\theta$, $b$, and $\alpha(1)$ (and to a lesser extent, $k$).

<table>
<thead>
<tr>
<th>$L^*$</th>
<th>$M = 2^{20}$, $b = 32$, $\alpha(1) = 100$</th>
<th>$M = 2^{25}$, $b = 16$, $\alpha(1) = 100$</th>
<th>$M = 2^{20}$, $b = 32$, $\alpha(1) = 100$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U(10)$</td>
<td>$U(100)$</td>
<td>$U(1000)$</td>
<td>$U(10)$</td>
</tr>
<tr>
<td>2044</td>
<td>1961</td>
<td>1952</td>
<td>1535</td>
</tr>
<tr>
<td>$(L^* - L^<em>)/L^</em>$</td>
<td>-0.022</td>
<td>0.02</td>
<td>0.024</td>
</tr>
<tr>
<td>$\log_2(L^<em>) - \log_2(L^</em>)$</td>
<td>-0.032</td>
<td>0.028</td>
<td>0.035</td>
</tr>
<tr>
<td>$\tilde{\alpha}^* = \alpha(L^<em>)/L^</em>$</td>
<td>0.066</td>
<td>0.068</td>
<td>0.069</td>
</tr>
<tr>
<td>$\text{Eff}(FS/OESk)$</td>
<td>0.72</td>
<td>0.72</td>
<td>0.72</td>
</tr>
</tbody>
</table>

**TABLE II**

Behavior of the optimum number of flow classes $L^*$, and OESk, for some TG models. From top to bottom: true $L^*$; relative error of $\hat{L}^*$; error in number of ownership bits predicted; conditional load at optimum (expect $L^*$). Efficiency compared to FS over $M/b$ counters. In each case $\alpha(1) = N_f/A(1) = bN_f/M = 100$.

<table>
<thead>
<tr>
<th>$L^*$</th>
<th>$M = 2^{20}$, $b = 32$, $\alpha(1) = 100$</th>
<th>$M = 2^{25}$, $b = 16$, $\alpha(1) = 100$</th>
<th>$M = 2^{20}$, $b = 32$, $\alpha(1) = 100$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U(10)$</td>
<td>$U(100)$</td>
<td>$U(1000)$</td>
<td>$U(10)$</td>
</tr>
<tr>
<td>2851</td>
<td>3831</td>
<td>4030</td>
<td>2207</td>
</tr>
<tr>
<td>$(L^* - L^<em>)/L^</em>$</td>
<td>-0.30</td>
<td>-0.48</td>
<td>-0.50</td>
</tr>
<tr>
<td>$\log_2(L^<em>) - \log_2(L^</em>)$</td>
<td>-0.51</td>
<td>-0.94</td>
<td>-1.0</td>
</tr>
<tr>
<td>$\tilde{\alpha}^* = \alpha(L^<em>)/L^</em>$</td>
<td>0.07</td>
<td>0.07</td>
<td>0.07</td>
</tr>
<tr>
<td>$\text{Eff}(FS/OESk)$</td>
<td>0.71</td>
<td>0.70</td>
<td>0.70</td>
</tr>
</tbody>
</table>

**TABLE III**

As for Table \(\text{II}\), but for $L^*_W$ (optimizing for $\theta_W$ instead of $\theta_1$). The optimum is achieved at higher $L$, and hence lower $p$. 
B. Some Implementation Considerations

Both methods are readily implementable. With only a single hash and a single counter array, FSS is very simple, cheap enough to implement alongside existing packet sampling based systems. By reusing the hash for priority class as well as counter assignment, ESK is also not onerous to implement [15], [16]. However, the need to look up, for each packet, the value of an ownership variable (and typically to modify it), is an additional per-packet cost which echoes in a lighter form the disadvantages of flow-table based methods which FSS avoids entirely. This cost grows with $\alpha$ as the resolution of the flow classes increases and the number of bits needed to record ownership tends toward that of a flow-id key.

As previously noted, the fact that $p_f$ is a free parameter, unrelated to memory cost, allows changes in the load $\alpha$ to be readily accommodated. When using OFSS, in view of Equation (23), one does not even have to reevaluate the underlying optimum controlling parameter $\alpha^*$. In contrast, for OESk not only will $L^*$ have to be reevaluated, a new value requires that the data structure be reconfigured to suit a new number $A(L)$ of counters and ownership variables with a revised number of bits. More generally, if the sampling rate provided by the method needs to be changed for any reason, this is trivial to perform in FSS, whereas for ESk the corresponding value of $L$ has to be derived (note $p < 1/\alpha$ is impossible), and the data structure reconfigured.

IX. Performance Comparisons

In this section we compare the CRLB performance of OFSS against that of two skampling competitors, SGS and ESk, and add additional comparisons to FS and Sk which are different to those given above. The comparisons here are based on a constant number $A$ of counters, rather than constant total memory $M$ as in (most of) Sections [VII and VII]. Although this gives ESk a memory advantage, and precludes a comparison against OESk, it allows a simpler comparison in an environment where $A$ and $\alpha$ are constants, and $b$ does not appear, enabling a wider exploration of the parameters $k$ and $\theta$. The comparisons against ESk are still of interest, as they are in a regime where $L$ is quite small and so the memory disparity can be overlooked to first order. The goal of this section is to gain a feeling for the methods in a more realistic concrete setting, and to examine SGS.

A. SGS

Sketch Guided Sampling, introduced in [11], is at heart a packet sampling method, in fact a family indexed by a sampling probability function $p(k)$, $k \in [W]$. A packet will be sampled with probability $p(k)$ if it is the $k$-th packet encountered so far in its flow. Of course, we do not know the in-flow position of packets, if one did $\theta$ would be already known! The innovation in [11] was to estimate this quantity by a coarse counter sketch, hence SGS is a skampling method.

To simplify comparisons, we assist SGS by replacing its sketch component with oracular knowledge of each packet’s in-flow position. Assisted SGS is then a pure sampling method, whose per-flow counter density is characterized by its sampling matrix $B$ as

$$c_j = \sum_{k=1}^{W} b_{jk} \theta_k, \quad 0 \leq j \leq W,$$

where $b_{jk}$ is the probability that if the original flow had $k$ packets, only $j$ remain after sampling. The matrix element $b_{jk}$ can be calculated via the recursion

$$b_{j,k+1} = p(k+1)b_{j-1,k} + q(k+1)b_{j,k}.$$  

The average packet sampling rate is given by

$$p = \frac{1}{D} \sum_{j=1}^{W} \sum_{k=1}^{W} j b_{jk} (\beta, \epsilon) \theta_k.$$  

From general sampling results from [20], we can write

$$J_{SGS} = B^T D B,$$  

where $D = \text{diag}(B \theta)$, and $J_{SGS}^+ = J_{SGS}^{-1} - \theta \theta^T$. Following [11], we use

$$p(k) = p(k; \beta, \epsilon) = \frac{1}{1 + \epsilon^2 k^{(2\beta - 1)}}, \quad 1/2 \leq \beta \leq 1,$$  

a monotonically decreasing function of $k$. The idea is that this creates a bias toward short flows which counteracts the strong large-flow bias of traditional i.i.d. packet sampling, corresponding to $p(k; 1/2, \epsilon) = p$, a constant.

More specifically, the form (43) of $p(k)$ was chosen as it ensures that the growth of the mean square error ($\ell_2$) with respect to flow size $k$ is bounded by $O(k^{2\beta - 1})$, which is linear in the case $\beta = 1$. As we see below however, this does not mean that its variance is bounded satisfactorily with respect to other parameters, in particular $\alpha$.

Note that (true) SGS suffers from flow-table related computational issues, as well as the costs of a sketch, and so is less favorable to high speed implementation than FS.

B. Basis of Comparison

We compare the CRLB performance of OFSS against that of FS, Sk, and enhanced versions of our skampling competitors SGS and ESk. For SGS($\beta, \epsilon$), we use the assisted form as described above, and also test using $\beta = 0.75$ in addition to the $\beta = 1$ value used in [11]. For ESk($L$), we compare not only against the $L = 2$ favored in [16], but also larger values adapted to $\alpha$, and ignore the additional memory required to maintain the ownership variables. In data estimation comparisons, we use Maximum Likelihood Estimators (MLEs) for all methods. Previously a sub-optimal estimator was used to test SGS and ESk.

We compare on the basis of equal number of counters. The methods Sk, FSS, OFSS, ESk are each given an array with $A$ counters, whereas the sampling methods FS, SGS are given flow-tables with $A$ entries. This means that if $N_f > A$, only $N_f^* = \min(N_f, A)$ flows will actually
be delivered to the methods (we ignore flow expiry). If this limitation were not imposed, then the variances would simply go to zero as $\alpha \to \infty$. For simplicity, for FS we achieve the target $N_f$ statistically, via the choice of $p_f$ given below, rather than deterministically.

With $A$ and $N_f$ (or equivalently $\alpha = N_f/A$) fixed, methods $S_k$, $ES_k(2)$ and $FS(p_f)$ are determined ($p_f = N_f'/N_f = \min(1,1/\alpha)$), as are $OFSS(k)$ given the target $\theta_k$. For $SGS$, for each of $\beta = \{0.75, 1\}$ we set $\varepsilon$ to match the average number of captured packets to that of FS. For $ES_k$, at larger $\alpha$ values we select two values of $L > 2$ to bracket $OFSS$ (at the chosen $k^*$). Finally, we also compare against $FS(1)$, that is perfect flow collection. (no resource constraints, all flows used with no distortion).

We compare over an operating range of $1 < \alpha \leq 100$, since $\alpha \approx 1$, though a small resource-rich value, is theoretically important, and $\alpha = 100$ corresponds to a commonly quoted sampling probability of $0.01$. Values as high as $\alpha = 10000$ are now typical in Big Data contexts. The structure of $OFSS$ is such that its performance will be unaffected by higher loads, whereas other methods suffer in information or implementation terms, or both, as load increases.

As before $A$ enters in only as a multiplicative factor, so we plot per-counter variance rather than total variance. Here the objective is to compare methods, the absolute CRLB values depend on $\theta$ and are not important.

C. CRLB Comparisons

We evaluate $(\mathcal{I}^+)^{kk}_{kk}$ numerically from the exact expressions given earlier, initially using a truncated geometric model $TG(W, R)$ for $\theta$, where $R = \theta_1/\theta_W$.

Consider Figure 9 where $\theta$ is uniform, $W = 50$. In plot(a) $\alpha \approx 1$, a load so light that results are close to degenerate: $OFSS(p_f^*) \approx FSS(1) = S_k \approx ES_k(2)$, $FS(p_f) \approx F(1)$. Despite this SGS already shows poor behavior. At $\alpha = 2$ it is already orders of magnitude worse than other methods for $k > 3$, and becomes almost impossible to calculate for $\alpha = 100$. This is consistent with the poor behavior of packet sampling [20], whose structure SGS generalizes, but does not fundamentally change. For small $\alpha$ $ES_k(2)$ outperforms $OFSS$, however as $\alpha$ increases both $Sk$ and $ES_k(2)$ also have variances orders of magnitude beyond $OFSS$ for all $k$, and then become difficult to calculate. At $\alpha = 100$ the field therefore narrows to $ES_k(L)$ for well chosen $L > 2$ versus $OFSS$. For each of plots (c) and (d) $L$ values are used so that $ES_k$ brackets $OFSS$. In plot(d) we see that, as expected, $OFSS$ is within a constant factor of, and the same order of magnitude as, the benchmark $FS(1/\alpha)$. We

<table>
<thead>
<tr>
<th>Trace</th>
<th>Link Capacity</th>
<th>$N_f$</th>
<th>Duration (hh:mm:ss)</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leipzig-II</td>
<td>50 Mbps</td>
<td>2,277,052</td>
<td>02:46:01</td>
<td>19.76</td>
</tr>
<tr>
<td>Abilene-III</td>
<td>10 Gbps</td>
<td>23,806,285</td>
<td>00:59:49</td>
<td>16.12</td>
</tr>
</tbody>
</table>

TABLE IV
SUMMARY OF THE DATA TRACES USED.
see that for ESk to (just) defeat OFSS requires $L = 64$. Assuming a generous $b = 32$ bits per counter, this means that $\approx 19\%$ extra memory is needed to achieve this, more for larger $\alpha$.

Comparing plot(d) with Figure 10(a) shows that similar conclusions hold even after changing the distribution shape quite radically from uniform to TG(50,1000). For TG($W, R$) with $R = \{10, 1000\}$, $W = \{50, 1000\}$, similar results were found. Changing from optimizing for $\theta_1$ to other $\theta_k$ (not shown) changes variance values somewhat for OFSS($k$), and the matching $L$ slightly for ESk($L$), but not the conclusions.

Figure 10 provides a bridge from models to data with load fixed at $\alpha = 100$. From plots(a) to (b) we increase $W$ from $W = 50$ to a more realistic $W = 1000$. The picture is remarkably unchanged. From plot(b) to (c) we move from a very rough traffic model, TG(1000,1000), to data from the Abilene-III dataset (see Table IV), truncated at $W = 1000$. Again the same model-comparison conclusions hold. Finally, plot(d) uses $\theta$ from the Leipzig-II dataset where $\theta_k = 0$ for many $k$, resulting in zeros manifesting as gaps in the FS and FS(1) curves. Associated ‘spiky’ far-tail estimates for OFSS and ESk are a sign of the need for truncation, here we used $W = 200$. Larger datasets will typically allow much longer truncations while still respecting the constraint $\theta_k > 0$ (Equation [1]).

D. Estimation Comparisons

We now compare $\hat{\theta}$ estimates, again for $\alpha = 100$, for FS, OFSS, SGS($1, \epsilon$), and ESk($64$) using maximum likelihood estimation (MLE derivations provided in Appendix D).

The datasets, summarized in Table IV, are old but adequate for testing the methods. We extract TCP flows according to the standard 5-tuple (with no timeout).

Figure 11 plots $\hat{\theta}$ for Abilene-III, truncated at $W = 2000$, which is approximately the largest value for which $\theta_k > 0$ for all $k$. The grey curve, FS(1), corresponds to $\theta$ itself. The estimate for SGS: $\hat{\theta}_1 = 1$, and $\hat{\theta}_k = 0$ for all $k > 1$, is as expected very poor, in fact degenerate. All other methods appear to perform quite well, however it is very difficult to assess performance reliably from such plots, in particular in the far tail because of the high variability inherent in single point estimates as the data ‘runs out’. Smoothing is typically used to improve behavior in such cases [4].

For a more objective assessment, we employ the $\ell_2$ error $||\hat{\theta} - \theta||_2 = (\sum_{k=1}^{W} (\hat{\theta}_k - \theta_k)^2)^{1/2}$ to summarize performance of each method over all $k$. Overall, for each trace the results of Table V reflect the variance pecking order FS<$\cdot$OFSS<$\cdot$ESk($64$)<SGS from the CRLB analysis averaged over all $k$, both when optimized for $\theta_1$ and $\theta_1\epsilon$. Four exceptions are noted in bold. In two of these OFSS is seen to defeat FS. This is due to an implicit smoothing of the sketch component of OFSS improving the fit over the tail where for many indices $\theta_k = 0$ for Leipzig. In the other two cases SGS appears comparable to ESk for Leipzig when $k^* = W$ instead of much worse. This results from the fact that $\theta_1, \theta_2$ are fairly large and SGS is able to give rough estimates for them given its deliberate small flow bias, while setting $\hat{\theta}_k = 0$ for all $k > 2$. Note that these exceptions are in part due to the limitations of $\ell_2$ as a summary metric. Leipzig-II provides an instructive example of difficulties (for all methods) which arise when

<table>
<thead>
<tr>
<th>Trace</th>
<th>$k^*$</th>
<th>$\epsilon_1$</th>
<th>$L$</th>
<th>FS</th>
<th>OFSS</th>
<th>ESk</th>
<th>SGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abilene</td>
<td>1</td>
<td>0.009</td>
<td>64</td>
<td>1.6e-3</td>
<td>5.6e-3</td>
<td>7.8e-2</td>
<td>8.9e-1</td>
</tr>
<tr>
<td>Leipzig</td>
<td>1</td>
<td>0.009</td>
<td>64</td>
<td>9.4e-2</td>
<td>2.1e-2</td>
<td>2.5e-1</td>
<td>2.8e-1</td>
</tr>
<tr>
<td>Abilene</td>
<td>W</td>
<td>0.005</td>
<td>64</td>
<td>1.6e-3</td>
<td>4.1e-3</td>
<td>7.8e-2</td>
<td>8.9e-1</td>
</tr>
<tr>
<td>Leipzig</td>
<td>W</td>
<td>0.013</td>
<td>64</td>
<td>9.4e-2</td>
<td>2.1e-2</td>
<td>2.5e-1</td>
<td>2.5e-1</td>
</tr>
</tbody>
</table>

TABLE V

$\ell_2$ ERROR: ABL.(W=2000), LEIP.(W=200).
the assumption of \( \theta_k > 0 \) is violated. Abilene-III is representative of a case where this is avoided by a suitable, and larger, choice of \( W \).

X. CONCLUSIONS

We have studied the problem of how to optimally collect information from packet stream data for the purpose of estimating the flow size distribution \( \theta \), quantified through the (constrained) Fisher Information.

Our first contributions lay in a detailed analysis of the information capacity of the counter array sketch \( S_k \). We proved results on the monotonic decrease of per-flow information with load \( \alpha \), and the large-\( \alpha \) structure of the eigenvalue spectrum of both the Fisher Information and the CRLB. This culminated in Theorem 7 which confirmed the key fact that overloading a sketch results in information destruction, and showed how the information carried could be maximized.

Motivated by the existence of an optimal sketch loading, we then defined a hybrid ‘skampling’ method, the Flow Sampled Sketch (FSS), which combines both sampling and sketching. The method allows the incoming flow rate to be reduced through a sampling parameter, \( p_f \), while retaining the very low computational character of the counter array. We derived a Fisher Information analysis of FSS, and proved in Theorem 28 that the optimum loading result for \( S_k \) can be used to yield an optimal setting \( p_f^* \) for FSS, which we then used to define the Optimized Flow Sampling Sketch (OFSS). Using a truncated geometric family of distributions and some others, we examined the behavior of \( p_f^* \) as a function of key parameters, and showed how to calculate it efficiently. We examined the efficiency of OFSS with respect to Flow Sampling, whose Fisher Information analysis had been reported in prior work \cite{20}, and found it to be approximately constant, as well as independent of \( \alpha \).

We then re-examined an existing skampling method, the Eviction Sketch (ESk) defined (but not named) in \cite{18}. We performed the first thorough Fisher Information analysis of ESK, and found that its information and variance bounds could be closely related to that of \( S_k \), and hence compared to OFSS in a natural way. We described how the free parameter of ESK, the number of flow classes \( L_k \), could be used to improve its performance, and showed how an optimal \( L_k^* \) could be defined under two different regimes: constant number of counters where \( L_k^* = \infty \), and constant total memory where \( L_k^* < \infty \). In the latter case we used \( L_k^* \) to define a second new method, the Optimized Eviction Sketch (OESk). We compared and contrasted ESk and FSS, and in more detail OFSS and OESk, from the points of view of efficiency, simplicity, ease of calibration, and in-principle implementability.

In summary, each of OFSS and OESk is a high efficiency method which is readily implementable even at the highest speeds. In practical and theoretical terms each solve the problem of how to engineer a method which is Flow-Sampling-like statistically, without using flow-tables. They therefore each inherit key advantages of FS, such as allowing good estimation over the entire \( \theta \) distribution, not only the tail (or head). OESk in particular, due to its deterministic ‘collision clearing’ ability, merits the title of an ‘implementable FS’. The advantages of OFSS are great simplicity, minimal implementation costs, and intuitive operation and parametrization, which leads to benefits such as trivial adaption to varying loads without any impact on statistical performance, processing time or memory use. The advantage of OESk is higher inherent efficiency due to stricter collision control. As \( \alpha \) grows however (which is to be expected if data volume growth exceeds hardware performance growth), the efficiency of OESk drops, and its implementation and operation moves toward that of flow-table methods.

Finally, Section 10 provides numerical comparisons of the performance of the methods as a function of \( \theta \) and \( \alpha \), using both models (CRLB evaluation) and actual Internet packet traces (CRLB evaluation as well as estimation). Here we added a comparison against another existing skampling method, Sketch Guided Sampling (SGS), for which we provide for the first time a Fisher Information analysis. We showed that the performance of SGS is extremely poor both in information and implementation terms, and so should not be used. For OFSS and suitably calibrated ESk the results were, as expected, very good compared against the Flow Sampling benchmark. For the estimation from data we derived Maximum Likelihood Estimators for each method (in most cases for the first time), in a fixed point implementation equivalent to a form of Expectation Maximization.

Our work is not limited to the context of core-router measurement. It is applicable to any situation where data consists of items that can be classified into groups (flows), and where either the items cannot be stored and so must be analyzed on the fly (stream processing), or when data sets are enormous and fast summaries over them are needed (Big Data).

ACKNOWLEDGMENT

Partially supported by Australian Research Council’s Discovery Projects funding schemes # DP120102834 and # DP110103505. The authors thank eResearch South Australia (eRSA) for the use of their computational resources.

APPENDIX

A. Generating Function of the FSS Counter Distribution

Let \( G_k(s) = 1 - \frac{1}{k + \frac{1}{4}s^k} \) be the generating function of a counter after a single flow of size \( k \in [W] \) is inserted into the sketch, and \( G_{M_k}(s) = (1 - p_f + p_f s)^{M_k} \) be that of the number of sampled flows of size \( k \), \(|s| < 1\). Using the independence of flows of different sizes, and the law of
total expectation, the generating function of $C$ is given by
\[
C^*(s) = E[\sum_{k=1}^W kX_k]
\]
\[
= E\left[E[\sum_{k=1}^W kX_k \mid M, M_1', \ldots, M_W']\right]
\]
\[
= E\left[E[\sum_{k=1}^W kX_k' \mid M_1'] \cdot \ldots \cdot E[\sum_{k=1}^W kX_k' \mid M_W']\right]
\]
\[
= E\left[E\left[G_1(s)M_1' \ldots G_W(s)M_W'\right] = G_M'(G_1(s)) \cdot \ldots \cdot G_M'(G_W(s))\right]
\]
\[
= \prod_{k=1}^W \left(1 - p_f + \frac{p_f(1 - 1/A + 1/A s^k)}{A}\right)^{M_k}
\]
which is the same form as for $S_k$, but with parameter scaled by $p_f$. The Poisson approximation is then $\prod_{k=1}^W \exp(-\frac{p_f}{A} s^k - 1) = e^{\sum k p_f s^k} = e^{\sum k\lambda_s s^k}$.

B. Fisher Information of $M$, # unsampled flows in FSS

The density of $M$ is $p_M(m) = \frac{\mu^m e^{-\mu}}{m!}$, $\mu = q_f N_f$. It is easy to show that $\partial f_j/\partial \theta_i = \partial \mu/\partial \theta_i (m/\mu - 1) = m - \mu$ since $\partial \mu/\partial \theta_k = \mu \forall k$. The corresponding Fisher Information is then
\[
[J_M]_{ij} = E[\partial \mu/\partial \theta_i \partial \mu/\partial \theta_j (m/\mu - 1)^2] = E[(m - \mu)^2] = \text{Var}[M] = \mu.
\]
The normalized per-flow information is $[J_M]_{ij} = \mu/N_f = q_f$. As this is the same for all $i, j$, we have $J_M = q_f W 1^T W$.

C. Fisher Information for ESK

The Fisher Information matrix for the joint observable $(O, C)$ is
\[
J(\theta) = \sum_{\ell=1}^L \sum_{j>0} (\nabla \theta \log w_{j,\ell})^T \nabla \theta \log w_{j,\ell}
\]
\[
= (\nabla \theta \log w_{0,L})^T (\nabla \theta \log w_{0,L}) + \sum_{\ell=1}^L (\nabla \theta \log w_{j,\ell})^T (\nabla \theta \log w_{j,\ell}).
\]
Set $f_j = c_{sk,j}(\hat{\alpha})$ and recall $E = e^{-\hat{\alpha} \sum \theta_k}$. We have $\partial E/\partial \theta_k = -\hat{\alpha} E$ and $\partial (\log E^{\ell-1})/\partial \theta_k = -\hat{\alpha}(\ell - 1)$. It follows that
\[
\frac{\partial \log w_{0,L}}{\partial \theta_k} = -\hat{\alpha} L
\]
\[
\frac{\partial \log w_{j,\ell}}{\partial \theta_k} = -\hat{\alpha}(\ell - 1) + \frac{\partial \log f_j}{\partial \theta_k}, \quad j > 0,
\]
resulting in
\[
(J)_{ik} = \hat{\alpha}^2 L^2 E^L + \sum_{\ell=1}^L \sum_{j>0} \left( -\hat{\alpha}(\ell - 1) + \frac{\partial \log f_j}{\partial \theta_k} \right) E^{\ell-1} f_j.
\]
The double sum expands into four terms.
\[
T_1 = \sum \sum_{\ell=1}^L \hat{\alpha}^2 (\ell - 1)^2 E^{\ell-1} f_j
\]
\[
= \hat{\alpha}^2 (1 - E) \sum_{\ell=1}^L E^{\ell-1} (\ell - 1)^2
\]
since $\sum_{j>0} f_j = 1 - f_0$, and $f_0 = e^{-\hat{\alpha}} = E$.
\[
T_2 = - \sum \sum_{j>0} \hat{\alpha}(\ell - 1) \frac{\partial \log f_j}{\partial \theta_k} E^{\ell-1} f_j
\]
\[
= -\hat{\alpha}^2 \sum_{\ell=1}^L E^{\ell-1}(\ell - 1) \sum_{j>0} (1_{j \geq k} f_{j-k} - f_j)
\]
\[
= -\hat{\alpha}^2 E \sum_{\ell=1}^L E^{\ell-1}(\ell - 1)
\]
using $[\prod]$, since the sum over $j > 0$ is just $f_0$. We also have $T_3 = T_2$ corresponding to $\partial/\partial \theta_i$. Finally
\[
T_4 = \sum \sum_{j>0} \hat{\alpha}(\ell - 1) \frac{\partial \log f_j}{\partial \theta_k} E^{\ell-1} f_j
\]
\[
= \sum_{\ell=1}^L E^{\ell-1} \sum_{j>0} \frac{\partial \log f_j}{\partial \theta_k} \frac{\partial \log f_j}{\partial \theta_i}
\]
\[
= F(L) \left( (J(S_k, \hat{\alpha}))_{ik} - \hat{\alpha}^2 E \right)
\]
\[
= F(L) \left( \hat{\alpha}(J(S_k, \hat{\alpha}))_{ik} - \hat{\alpha}^2 E \right)
\]
since $\partial \log f_0/\partial \theta_k = -\hat{\alpha}$, so the missing term is $\hat{\alpha}^2 f_0$.
We now have
\[
(J)_{ik} = \hat{\alpha}^2 L^2 E^L + T_1 + T_2 + T_3 + T_4
\]
\[
= \hat{\alpha}^2 H + F(L) \hat{\alpha}(J(S_k, \hat{\alpha}))_{ik}
\]
as claimed, where $H = L^2 E^L + (T_1 + 2T_2)/\hat{\alpha}^2 - F(L) E$ depends only on $\alpha$ and $L$. Note that $L = 1$ implies $F = 1$, $H = 0$, and $\hat{\alpha} = \alpha$ and we recover $J_{sk}(\alpha)$ as expected.
To simplify $H$, let $F_i = \sum_{\ell=1}^L E^{\ell-1} \hat{\alpha}^i$ (hence $F = F_0$).

Then $H = L^2 E^L + F_0 - 2F_1 + (1 - E)F_2$.

D. MLE for sampling and sketching methods

The likelihood function for $N_f$ flows is given by
\[
f(\theta, N_f) = \prod_i f(j_i; \theta) = \prod_{j=0}^{N_f} c_j(\theta)^{M_j'}
\]
where $M_j'$ is the number of counters recording $j$ packets. The MLE is the $\theta$ which maximizes the log-likelihood subject to the constraint $\sum_{k=1}^W \theta_k = 1$, $\theta_k > 0$, $\forall k$. The corresponding Lagrangian is given by
\[
\mathcal{L}(\theta, \mu, \nu) = \sum_{j=0}^{N_f} M_j c_j(\theta) - \mu (\sum_{k=1}^W \theta_k - 1) - \nu^T \theta,
\]
where the vector $\nu$ has elements $\nu_k \geq 0$ and $\mu \in \mathbb{R}$. By differentiating with respect to $\theta_k, \forall k$ and the multipliers,

$$\frac{\partial L(\theta, \mu, \nu)}{\partial \theta_k} = \sum_{j=k}^{\infty} M_j^0 \frac{\partial c_j(\theta)}{\partial \theta_k} - \mu - \nu_k = 0, \quad (50)$$

$$\frac{\partial L(\theta, \mu, \nu)}{\partial \mu} = 1 - \sum_{k=1}^{W} \theta_k = 0,$$

$$\frac{\partial L(\theta, \mu, \nu)}{\partial \nu_k} = \theta_k = 0.$$

The second equation is just the equality constraint. The third implies $\theta = 0_W$ which contradicts $0 < \theta_k < 1$ for all $k$, so the Karush-Kuhn-Tucker condition implies that $\nu = 0_W$.

All the MLEs are solutions to fixed point equations (with slight differences depending on the method). The solutions are obtained by iterating $\theta_{n+1} = \Theta(\theta_n)$ until convergence starting from the initial condition

$$\hat{\theta}^{(\text{init})} = \frac{1}{\sum_{j=1}^{W} M_j'} [M_1', M_2', \ldots, M_W']^T. \quad (51)$$

We now separate the discussion into the sampling and sketching/skampling cases as the nature of the counter density is quite different.

1) **MLE for Sampling**: In [20] we derived the MLE for a class of bias-free methods, which in the case of flow sampling reduced to

$$\hat{\theta} = \frac{1}{\sum_{j=1}^{W} M_j'} [M_1', M_2', \ldots, M_W']^T. \quad (52)$$

This is an explicit expression which is very simple to evaluate. In the general case matrix inversion is required. We now derive a fixed point approach that works for arbitrary $B$ without the need for matrix inversion. We use this method to evaluate the MLE for SGS.

Recall that for sampling methods characterizable by a sampling matrix $B$ the counter density is $c(\theta) = B\theta$. It follows that (50) (with $\nu_k$ set to zero) can be written in matrix form as

$$B^T \text{diag} \left( \frac{M_0^0}{c_0(\theta)}, \frac{M_1'}{c_1(\theta)}, \ldots, \frac{M_W'}{c_W(\theta)} \right) \mathbf{1}_{W+1} = \mu \mathbf{1}_W, \quad (53)$$

which after multiplying both sides by diag($\theta$) becomes

$$\text{diag}(\theta) B^T \text{diag} \left( \frac{M_0^0}{c_0(\theta)}, \frac{M_1'}{c_1(\theta)}, \ldots, \frac{M_W'}{c_W(\theta)} \right) \mathbf{1}_{W+1} = \mu \mathbf{1}_W.$$

By multiplying both sides by $\mathbf{1}_W^T$ we find $\mu = N_f$. Thus

$$\hat{\theta} = \frac{\text{diag}(\hat{\theta})}{N_f} B^T \text{diag} \left( \frac{M_0^0}{c_0(\theta)}, \frac{M_1'}{c_1(\theta)}, \ldots, \frac{M_W'}{c_W(\theta)} \right) \mathbf{1}_{W+1}, \quad (54)$$

where the MLE $\hat{\theta}$ is the solution to this fixed point equation, which we write $\hat{\theta} = \Theta(\hat{\theta})$ where $\Theta$ is the function appearing on the right hand side.

**EM formulation.** The fixed point MLE formulation can be interpreted in terms of the EM algorithm. The algorithm is divided to two iterative steps: the *Expectation* (E-) and *Maximization* (M-) steps, and successive refinement of the parameter estimates are made by alternating between these two steps.

In a sense the EM algorithm works by trying to estimate information about $\theta$ that is missing from the observations but which, if available, would make estimation simple. Here the missing information are the $M_k$s. To aid in the estimation of $M_k$, we estimate the probability $p(k; j, \theta)$ that an observed sample of $C = j$ packets came from a flow of size $k$. Clearly

$$p(k; j, \theta) = \frac{b_{j,k} \theta_k}{\sum_{K=1}^{W} b_{j,K} \theta_K}.$$

Since this is not observable, the E-step is given by the minimum mean squared error estimate (MMSE) of each $M_k$ given the samples $M'_j, \forall j$, conditioned on an estimate $\hat{\theta}$ of $\theta$. For each $k$ it is defined as

$$\hat{M}_k = \frac{\sum_{j=0}^{W} b_{j,k} \theta_k}{\sum_{j=1}^{W} M_j'}$$

Note that $\sum_{j=1}^{W} \hat{M}_j = \sum_{j=0}^{W} M_j' = N_f$.

By using Lagrange multipliers, it is easy to derive the M-step which is

$$\hat{\theta}_k = \frac{\hat{M}_k}{\sum_{j=1}^{W} M_j'} = \frac{\hat{M}_k}{N_f}.$$


To show how (54) is equivalent to the EM formulation, we re-express the right hand side of (54) as

$$\hat{\theta} = \frac{1}{N_f} \sum_{j=0}^{W} b_{j,\theta_k} \hat{M}_j' \left( \frac{b_{j,\theta_k} M_j'}{\sum_{K=1}^{W} b_{j,K} \theta_K} \right).$$

Since $c_j(\hat{\theta}) = b_{j,\theta_k} \hat{M}_j'$, each entry in the vector is equal to equation (55) (the E-step), divided by $N_f$ (the M-step). Thus, the fixed point algorithm combines both the E- and M-steps in a single iteration.

2) **MLE for Sketching**: We rewrite equation (50) (with $\nu_k$ set to zero) as

$$\frac{\partial L(\theta, \mu, \nu)}{\partial \theta_k} = \sum_{j=0}^{\infty} M_j' \frac{\partial \log c_j(\theta)}{\partial \theta_k} - \mu = 0. \quad (56)$$

This equation will be the starting point for each of the cases below. Recall from Theorem [1] that

$$\frac{\partial c_{Sk,j}(\alpha)}{\partial \theta_k} = \begin{cases} \alpha c_{Sk,j-k}(\alpha) - \alpha c_{Sk,j}(\alpha), & \text{if } j \geq k, \\ -\alpha c_{Sk,j}(\alpha), & \text{otherwise}. \end{cases} \quad (57)$$

**Sk.** Using (57) (here $c_j = c_{Sk,j}$), (56) becomes

$$\frac{\partial L(\theta, \mu, \nu)}{\partial \theta_k} = \alpha \sum_{j=0}^{\infty} c_{j-k}(\theta) M_j' - \alpha \sum_{j=0}^{\infty} M_j' - \mu = 0. \quad (58)$$
We then have, \( \forall k \),
\[
\alpha \sum_{j=1}^{W} c_{j-k} \left( \frac{1}{c_{j}((\hat{\theta})_{j})} \right) M'_{j} = \alpha a + \mu
\]  
(59)
since \( \sum_{j=0}^{\infty} M'_{j} = a \). Multiplying both sides by \( \hat{\theta}_{k} \):
\[
\alpha \hat{\theta}_{k} \sum_{j=k}^{W} c_{j-k} \left( \frac{1}{c_{j}((\hat{\theta})_{j})} \right) M'_{j} = (\alpha a + \mu) \hat{\theta}_{k}.
\]  
(60)
Summing over all \( \theta_{k} \), and using \( \sum_{k=1}^{W} \theta_{k} = 1 \), we get
\[
\mu = \alpha \sum_{k=1}^{W} \hat{\theta}_{k} \sum_{j=k}^{W} c_{j-k} \left( \frac{1}{c_{j}((\hat{\theta})_{j})} \right) M'_{j} - \alpha a.
\]
Plugging this term back into (60) yields
\[
\hat{\theta}_{k} \sum_{j=k}^{W} c_{j-k} \left( \frac{1}{c_{j}((\hat{\theta})_{j})} \right) M'_{j} = \hat{\theta}_{k} \sum_{\ell=1}^{W} \hat{\theta}_{\ell} \sum_{j=\ell}^{W} c_{j-\ell} \left( \frac{1}{c_{j}((\hat{\theta})_{j})} \right) M'_{j}, \ \forall k.
\]  
(61)
which we can express as a fixed point equation with
\[
\hat{\theta}_{k} = \frac{\hat{\theta}_{k} \sum_{j=k}^{W} c_{j-k} \left( \frac{1}{c_{j}((\hat{\theta})_{j})} \right) M'_{j}}{\sum_{\ell=1}^{W} \hat{\theta}_{\ell} \sum_{j=\ell}^{W} c_{j-\ell} \left( \frac{1}{c_{j}((\hat{\theta})_{j})} \right) M'_{j}}.
\]  
(62)

**FSS and OFSS.** For FSS recall that \( c_{j} = c_{sk,j}(\alpha') \). Since \( \alpha' = p_{ij} \alpha \) has no dependence on \( \theta \), the fixed point equation (62) applies on replacing \( c_{j} \) with \( c_{sk,j}(\alpha') \). For OFSS \( p_{ij} \) depends on \( \theta \) via \( \alpha' \), however operationally OFSS is based on a preselected \( p_{ij} \) which is fixed during (at least) a measurement interval. Thus \( \alpha' \) is fixed and again equation (62) applies.

**ESk.** As discussed in Section [X], we only need to consider the marginal density \( c_{j} \) of packet counts, rather than the joint density of ESk. Recall from Appendix [C] that \( f_{j}(\theta) = c_{sk,j}(\hat{\alpha}) \), and that for \( j > 0 \)
\[
\log c_{j}(\theta) = \log f_{j}(\theta) + \log F(L) = \log f_{j}(\theta) + \log(1 - E_{L}) - \log(1 - E).
\]
Hence, for \( j > 0 \), using equation (57)
\[
\frac{\partial \log c_{j}(\theta)}{\partial \theta_{k}} = \hat{\alpha} \left( \frac{f_{j-k}(\theta)}{f_{j}(\theta)} - 1 \right) + \frac{\alpha E_{L}}{1 - E_{L} - 1 - E}.
\]
and for \( j = 0 \), since \( c_{0}(\theta) = E_{L} \),
\[
\frac{\partial \log c_{0}(\theta)}{\partial \theta_{k}} = -\alpha.
\]
Using the above relations and equation (50) yields (63). Note that \( \sum_{j=0}^{\infty} M'_{j} = a \), so \( \sum_{j=0}^{\infty} M'_{j} = a - M_{0} \). Simplifying (63) and multiplying both sides by \( \theta_{k} \) yields equation (64). To solve for \( \mu \), we sum over all \( \theta_{k} \) and use the equality constraint to obtain \( E = e^{-\alpha} \) and \( E_{L} = e^{-\alpha} \), resulting in the expression (65) for \( \mu \). Substituting back into (64), many terms cancel to obtain
\[
\hat{\theta}_{k} \sum_{j=k}^{W} f_{j-k}(\hat{\theta}) M'_{j} = \hat{\theta}_{k} \sum_{\ell=1}^{W} \hat{\theta}_{\ell} \sum_{j=\ell}^{W} f_{j-\ell}(\hat{\theta}) M'_{j}.
\]
By noting that \( \frac{f_{j-k}(\hat{\theta})}{f_{j}(\hat{\theta})} = \frac{F(L) f_{j-k}(\hat{\theta})}{F(L) f_{j}(\hat{\theta})} = c_{j-k}(\hat{\theta}) \), once again one obtains a fixed point equation of the form (62).

**EM formulation.** The procedure is outlined in Algorithm [2] and was proven to be an instance of the EM algorithm in [9] for Sk.
the procedure is similar to using the law of large numbers to estimate the proportion of counters that came from a particular pattern $x \in \Omega_j$. As before, if we knew $M_k$, then recovering $\hat{\theta}$ would be simple, and the purpose of the E-step is to estimate this quantity for each $k$. The function (line 8 in Algorithm 2)

$$Q(\hat{M}_k; \hat{\theta}) := \sum_{x \in \Omega_j} x_k p(x, j, \hat{\theta}) M_j'$$

is the MMSE estimate of the $\hat{M}_k$. In the M-step, the algorithm finds the $\hat{\theta}$ that will optimize $Q(\hat{M}_k; \hat{\theta})$ based on the current estimate of $\hat{M}_k$. The update rule given in the M-step (line 12), which can be proven using Lagrangian multipliers [3], consists of simply taking the current empirical histogram of the estimated flow sizes. The algorithm iterates between these two steps until convergence.

We now show that the fixed point equation (61) is equivalent to Algorithm 2. To go any further, we need to develop an understanding of the left hand side of (61). Let $x_k$ denote the number of flows of size $k$, $x = [x_1, \ldots, x_W]$ be a flow collision pattern and $\Omega_j$ be the set of flow collision patterns with packet count $j$.

For a particular flow collision pattern $x$,

$$\Pr(X = x) = e^{-\alpha} \prod_{k=1}^{W} \frac{\lambda^x_k}{x_k!}.$$ 

Let $p(x, j, \theta)$ be the conditional probability of a particular collision pattern $x$ out of the set $\Omega_j$. Then,

$$p(x, j, \theta) := \Pr(X = x | C = j) = \frac{\Pr(X = x) \Pr(C = j | X = x)}{\sum_{x \in \Omega_j} \Pr(X = x) \Pr(C = j | X = x)}$$

$$= \frac{\Pr(X = x) \Pr(C = j | X = x) \prod_{k=1}^{W} \frac{\lambda^x_k}{x_k!}}{\sum_{x \in \Omega_j} \Pr(X = x) \prod_{k=1}^{W} \frac{\lambda^x_k}{x_k!}}$$

since $\Pr(C = j | X = x) = 1$ if and only if $x \in \Omega_j$.

We now prove that for $k \in [W]$

$$\alpha \theta_k \frac{c_{j-k}}{c_j} M_j' = \sum_{x \in \Omega_j} x_k p(x, j, \theta) M_j'.$$

(67)

Let $\Omega_{j,k}$ be the set of collision patterns of flows that sum up to packet count $j$ conditioned on $x_k \geq 1$. Then,

$$\alpha \theta_k \frac{c_{j-k}}{c_j} M_j' = \frac{\lambda_k}{\sum_{x \in \Omega_j} \prod_{\ell=1}^{W} \frac{\lambda_{x_k}^{\ell}}{x_k!}} M_j'$$

$$= \sum_{x \in \Omega_j} \prod_{\ell=1}^{W} \frac{\lambda_{x_k}^{\ell}}{x_k!} M_j'$$

$$= \sum_{x \in \Omega_j} \prod_{\ell=1}^{W} \frac{\lambda_{x_k}^{\ell}}{x_k!} M_j'$$

$$= \sum_{x \in \Omega_j} \prod_{\ell=1}^{W} \frac{\lambda_{x_k}^{\ell}}{x_k!} M_j'$$

(68)

In (68), we can rewrite the equation via a change of variables, since the numerator is equivalent to the probability of a collision pattern $x \in \Omega_{j,k}$ where $x_k \geq 1$. The last line follows from the fact that collision patterns with $x_k = 0$ will not contribute to the sum. Essentially, this is an estimate of the number of flows of size $k$, hence,

$$\hat{M}_k = \sum_{x \in \Omega_j} x_k p(x, j, \theta) M_j'.$$

(69)

In (69), each $M_j'$ is broken down according to the different possible collision patterns weighted by their probabilities. For example, suppose $M_3' = 1000$ counters. There are three possible patterns in this case: $x_3 = 1$, $x_2 = 1$, $x_1 = 1$, or $x_1 = 3$. If the respective probabilities for these patterns are 0.6, 0.3 and 0.1, then 600, 300 and 100 counters are estimated to belong to the first, second and third patterns respectively. The algorithm then estimates that there are 600 flows of size 3, 300 flows of size 2, and 600 flows of size 1, a total of 1500 flows arising from the observation of $M_3'$.

As the MLE derivations in the previous sections show, Algorithm 2 applies to FSS, OFSS and ESK simply by using load $\alpha'$, $\alpha'$ and $\alpha$ respectively. The interpretation is similar to the above for Sk, but instead of estimating the input $M_k$, since there are discarded flows in FSS and ESK, the algorithm estimates the ‘effective $M_k$’, the number of flows of size $k$ actually in the counters.
References


Paul Tune (M’09) received his PhD in Electrical and Electronics Engineering in 2010 from the University of Melbourne, Australia. He is now a Postdoctoral Research Fellow in the School of Mathematical Sciences at the University of Adelaide, Australia. His research interests are on Internet traffic matrices, random graphs, information theory, statistics and signal processing, specifically sparse signal recovery.

Darryl Veitch (F’10) completed a BSc. Hons. at Monash University, Australia (1985) and a mathematics Ph.D. from Cambridge, England (1990). He worked at TRL (Telstra, Melbourne), CNET (France Telecom, Paris), KTH (Stockholm), INRIA (Sophia Antipolis, France), Bellcore (New Jersey), RMIT (Melbourne) and EMIlab and CUBIN at The University of Melbourne, where he is a Professor Research Fellow. His research interests are in computer networking and include traffic modelling, parameter estimation, the theory and practice of active measurement, traffic sampling and sketching, and clock synchronisation over networks. He is a Fellow of the IEEE.