Fair Sampling Across Network Flow Measurements

Nick Duffield
AT&T Labs–Research
180 Park Avenue
Florham Park, NJ 07932, USA
duffield@research.att.com

ABSTRACT

Sampling is crucial for controlling resource consumption by internet traffic flow measurements. Routers use Packet Sampled NetFlow [9], and completed flow records are sampled in the measurement infrastructure [13]. Recent research, motivated by the need of service providers to accurately measure both small and large traffic subpopulations, has focused on distributing a packet sampling budget amongst subpopulations [26, 33, 40]. But long timescales of hardware development and lower bandwidth costs motivate post-measurement analysis of complete flow records at collectors instead. Sampling in collector databases then manages data volumes, yielding general purpose summaries that are rapidly queried to trigger drill-down analysis on a time limited window of full data. These are sufficiently small to be archived.

This paper addresses the problem of distributing a sampling budget over subpopulations of flow records. Estimation accuracy goals are met by fairly sharing the budget. We establish a correspondence between the type of accuracy goal, and the flavor of fair sharing used. A streaming Max-Min Fair Sampling algorithm fairly shares the sampling budget across subpopulations, with sampling as a mechanism to deallocate budget. This provides timely samples and is robust against uncertainties in configuration and demand. We illustrate using flow records from an access router of a large ISP, where rates over interface traffic subpopulations vary over several orders of magnitude. We detail an implementation whose computational cost is no worse than subpopulation-oblivious sampling.

Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks**]: Network Operations—*Network Monitoring*; G.3 [**Probability and Statistics**]: Probabilistic Algorithms

General Terms

Algorithms, Measurement, Theory

Keywords

Sampling, Estimation, Max-Min Fairness, IP Flows, Streaming

1. INTRODUCTION

1.1 Flow Records and Network Management

Passive traffic flow measurement plays a vital role in managing communications networks [20]. Flow measurement is performed by routers, most prominently by the NetFlow [9] feature, and more flexibly by standalone measurement devices [11] observing traffic via port replication or optical splitters. The measurement data are flow records, each of which summarizes the protocol-level information of a set of packets related by having a common key—comprising IP address and TCP/UDP ports and other header fields—observed locally in time. A flow record reports the flow key and other quantities, e.g., the flow's number of packets and bytes, and timing information. Each measuring device exports a stream of flow records—either directly or through co-located mediation devices—to one or more collectors that house a database whose functions include routine reporting (e.g., large scale traffic matrices and application mix) real-time anomaly detection (of traffic patterns due to network attacks or routing anomalies), and ad hoc retrospective queries (for forensic studies or debugging); see e.g., [20, 12].

1.2 Measurement Costs and Summarization

Several architectures for flow record collection and analysis currently are used, with differences according to relative situational costs of transmission, storage, and processing the flow records. The volume of flow records is immense; a major ISP will carry 10's of PetaBytes of user traffic per day [1], potentially generating 100's of TeraBytes of flow records daily. Intelligent sampling has played, and, we argue, will continue to play, an essential role in managing this data volume. In recent practice, sampling functionality has been deployed at or close to the measurement point, with sampling of packets in routers prior to the formation of flow statistics [9], then sampling of completed flow records in mediation devices [13] that export the sampled flow records to a central collector.

The bandwidth for 100 TB of flow records daily is only about 10Gb/s. The combined effects of decreasing bandwidth costs [29] and the advent of relatively cheap standalone measurement devices that need not packet sample, coupled with the increasing needs of network management for protocol-level measurement of the complete traffic stream, have made the collection of full sets of unsampled flow records

SIGMETRICS'12, June 11–15, 2012, London, England, UK.
Copyright 2012 ACM 978-1-4503-1097-0/12/06 ...$10.00.

increasingly feasible and desirable for ISPs. Although much ingenuity has been devoted to packet-level approaches to measurement resource management [26, 33, 40], the relatively long development times of hardware features makes full collection of flow records a flexible and compelling approach for ISPs, since development of analysis capabilities on the collector side is both flexible and rapid.

However, the volume of flow records remains a challenge. While retaining flow records over a limited time window (e.g., one or few days) is feasible in terms of storage cost, it remains problematic both to store all flow records over longer periods and to rapidly process arbitrary database queries over even the limited time window. A solution to both these problems is to summarize the flow records at collectors through a combination of aggregation and sampling. Although summarization incurs a loss of resolution for analysis, the reduced data size enables more rapid automated identification of network events large enough to be reliably detected, potentially triggering (fast) drill-down analysis into the summary, or (slower) drill-down into the full set of flow records available within the retention window. Furthermore, the summaries can be retained to support fast retrospective analysis over longer periods than it is feasible to retain the full flow data[1]. This architecture can be realized either entirely in a central collector, or in distributed local collectors. These routinely transmit summaries to a central location for global analysis and can initiate or remotely service queries on the limited window of full flow records that they store. One hundred such collectors, each equipped with a few TeraBytes of storage, could collectively retain a day's flow records from an entire major ISP network.

Aggregation and sampling are two complementary methods of summarization. Aggregation involves partitioning the stream of flow records into disjoint *subpopulations* (SPs) according to some explicit or derived property, and computing the aggregate flow, byte and/or packet counts within each SP over successive time windows. Multiple aggregation schemes may be run in parallel, e.g., according to single fields of the flow key, such as (i) source or destination IP routing prefix; (ii) source or destination TCP/UDP port; or (iii) router interface traversed. Time series of aggregates are useful both for routine reporting and anomaly detection. But general drill-down queries concern subaggregates over finer SPs of flows, and it is not feasible to compute and store all possible subaggregates of interest due to the high cardinality of the key space. Instead, sampling provides a data summary unconstrained by the granularity of any flow key aggregation scheme, and supports estimation of arbitrary subaggregates that need not specified at the time that sampling takes place.

1.3 The Challenge of Rate Skewness

We abstract the space constraints to which any summary must conform (either for storage or rapid query) as a sampling budget, i.e., a maximum number of samples to be gathered from data arriving in a specified time period. A feasible sampling method for flow records must (i) control sample size within a sampling budget; (ii) yield sufficiently accurate estimates; and (iii) provide samples in a sufficiently timely manner for applications. In particular, the sample design must match the statistical distribution of flow features to the relevant accuracy measures for estimation. If a numerical data feature has a heavily skewed distribution, such as a Pareto or other heavy-tailed distribution, and is used for selection and/or aggregation by a query, then it is well-known that estimation from uniformly drawn samples can be highly inaccurate [7]. An example comes from the measured distribution of flow sizes in which a small fraction of the flow records report a large fraction of the bytes [19]. While uniform sampling is well adapted to estimating subpopulation sizes (i.e., a number of flows in a subset), it is poorly adapted to estimating subpopulation weight sums (i.e., the total bytes of a subset of flows), since omission of a single large flow can drastically impact estimation accuracy. Instead, weighted sampling to preferentially select records of large flows [14, 16] enables accurate byte estimation.

This paper addresses skewness of a different type, that of the distribution of traffic rates over SPs. Drill-down queries often are localized within a SP of flow records whose aggregate timeseries triggered an anomaly. In many cases of interest, the traffic rates across SPs are highly skewed. Mirroring the general observation cited above, the basic problem that we address is that estimation accuracy varies greatly across SPs when sampling is *undifferentiated* in the sense that a flow record's sampling probability does not depend on the SP of which it is a member, although it may still depend on other attributes such as flow size.

Two examples of SP rate skewness are as follows. Firstly, when flow records are partitioned according to TCP/UDP port, a small number of ports account for a majority of the traffic, due to the relative popularity of different user applications [4]. Secondly, for analysis of traffic at an access router, flow records are implicitly partitioned according to the interface that the flow traversed, in order to analyze different customers' traffic. But line rates and corresponding traffic rates can vary by several orders of magnitude over different interfaces on a large access router. In both examples, the challenge for analysis is to return accurate responses to queries on traffic in each SP regardless of the SP size. For example, traffic monitoring for security must be able to alert on and drill-down into significant traffic changes on both small and large customer interfaces.

A simple analysis is illustrative. Consider a SP comprising n flow records which are then sampled independently with probability p. Now estimate from the samples the number m of flows in some subset of flows within the population. Whereas a "needle in a haystack" query might target a subset of size m within any SP, queries often concern SPs whose size m increases with the size n of the SP of which they are part, e.g., application-level flows comprising a fraction of all flows on each interface of an access router. The relative error in estimating m is roughly $1/\sqrt{mp}$; see eq. (4) following. With fixed p, estimation accuracy decreases for smaller m of typical interest in smaller n SPs. But if the sampling probability p were differentiated by SP size n, it would be possible to make accuracy uniform in n, redistributing some of the sampling budget to SPs with small n.

1.4 Optimally Sharing The Sampling Budget

Moving beyond the example analyzed above, consider the following static problem. Given a set of SPs of flow records collected in a time window, and a sampling methodology (e.g., flow size weighted), allocate a fixed total sampling budget between the SPs in order to realize certain estima-

[1] The analytic value of full data over summaries decreases with age, which may in any case be curtailed by policy.

tion accuracy goals for subsets of traffic within each SP. This budget allocation problem can be formulated in terms of fair sharing: the allocation to any SP will depend not only on its own flow records, but also on the demands of flow records in other SPs. Each SP needs to receive sufficient budget to realize its own accuracy goals; once these are met, any remaining sampling budget can be allocated to other SPs that can use them. Several variants of fairness are in general use, the best known being Max-Min Fairness and Proportional Fairness; see [2, 24]. Which one should be used? In this paper we will see that the type of accuracy measure employed determines which variant of fairness should be used. Put another way, fair sharing of sampling resources is motivated by the need to realize specific accuracy goals.

A sampling scheme using this approach could be used offline, batches of flow records being collected over successive measurement windows, and each SP sampled at the end of each window to fit within its fair share of the budget. However the lag between collection and summarization, being the sampling processing time plus an amount as large as the window duration, may be problematic for measurement applications that must query summaries in real time. This constraint favors online sampling in order to make samples available at any time. Some type of reservoir sampling [39] is well suited to this task: a reservoir of samples is progressively maintained throughout each window and is hence available for query at any time. Each arriving flow record must either be included in the reservoir (displacing a previous sample from that window if the reservoir is full) or discarded permanently. Since the sampling budget is the reservoir size, reservoir sampling also provides (i) robustness against unexpectedly large sets of flow records that would otherwise overflow storage, and (ii) the flexibility to sample flow records at devices with limited storage budget.

The online scheme should not simply reservoir sample with some fixed allocation of sampling budget amongst the SPs, e.g., based on modeled or predicted flow record arrivals, since these allocations would not in general be fair across the actual flow record arrivals, and hence suboptimal for estimation accuracy. Furthermore, the set of SPs represented in the flow records (e.g., which TCP/UDP ports occur) may not be known in advance. Instead, the online scheme must adapt the allocation in response to the actual flows records. Those that are sampled may subsequently be discarded, either to make way for preferentially sampled records in the same SP, or if the allocation to that SP is reduced.

1.5 Contribution: Fair Sampling

In our setting, there is a population of items (e.g., flow records) each with a weight (flow bytes, packets or count), and belonging to one set of subpopulations. Items are sampled into a reservoir of finite capacity (the sampling budget), and the sampled items are used to construct unbiased estimators of weight sums over subsets within SPs. The population is presented as a stream; each item must be either selected into the reservoir or discarded permanently. Selected items may later be discarded to make way for newer items. A high-level statement of our problem is then: *How should a sampling scheme share the sampling budget amongst* SP*s in order to best fulfill desired goals in estimation accuracy?*

Our proposed method of *Fair Sampling* integrates ideas from fair sharing, sampling and complexity analysis to solve this problem in a provably optimal and demonstrably feasible manner. We now outline the content of the paper and, in the list below, our main new contributions. We start in Section 2 by reviewing the notions of fair allocation and the sampling methodologies that we will employ.

- ◦ Section 3 relates different types of estimation accuracy to the type of fair sharing used in an offline context. Proportional Fair sharing minimizes the average estimation variance over SPs, and achieves the same accuracy as Undifferentiated Sampling (i.e., sampling items without regard to their SP membership). Max-Min Fair Sharing minimizes average *relative* estimation variance over SPs, and the resulting Fair Sampling increases accuracy for smaller SPs at the expense of decreasing accuracy for larger SPs.
- ◦ Section 4.1 presents a Streaming Fair Allocation algorithm that progressively and fairly allocates the sampling budget amongst SPs, without advance knowledge of their demand. The allocation is Lexicographic Max-Min Fair, which inherently accommodates the discrete nature of flow records.
- ◦ Section 4.2 presents a Streaming Fair Sampling algorithm that integrates sampling as a discard step when the Streaming Fair Allocation to some SP must be decreased. The VarOpt sampling algorithm [10] is variance optimal for estimation and can be implemented efficiently.
- ◦ Section 5 describes an implementation of Fair Sampling in a single shared reservoir whose computational cost is no worse than Undifferentiated VarOpt Sampling, i.e., $O(\log k)$ per item for a reservoir of capacity k.
- ◦ Section 6 uses flow records of traffic at an access router to evaluate the accuracy of Fair Sampling relative to other approaches, and its composite with Undifferentiated Sampling. It improves estimation accuracy in 89% of cases.
- ◦ Section 7 analyzes the performance gains of Fair Sampling, and provides measurement system design criteria that include the effects of any prior packet sampling.

We discuss related work in Section 8 and conclude in Section 9. Theorems in Sections 3 & 4 are proved in Section 10.

2. REVIEW: FAIRNESS AND SAMPLING

Before presenting our contribution in Sections 3 and following, we review the notions of budget sharing that underpin our work (Section 2.1), motivation in sampling and estimation (Section 2.2), the specific sampling and estimation algorithms that we employ (Section 2.3), and how different sample sets—in our case from Fair and Undifferentiated Sampling—can be combined into a single estimator (Section 2.4). Our setting employs the following notation that is retained throughout the paper. We assume a population of n items, partitioned in disjoint subpopulations[2] (SPs) labelled by $d \in \mathcal{D}$, there being n_d items in SP d, hence $\sum_d n_d = n$. The sampling budget k is the number of items that can be sampled. Each subpopulation d is allocated some non-negative integer amount $k_d \leq n_d$ such that $\sum_d k_d \leq k$; such allocations are called *feasible*. Lastly, each item i comes equipped a weight $w_i > 0$ that determines its sampling probability within each SP. For the motivating case where items are flow records, examples of subpopulation partitions were mentioned in Section 1.3; the weights

[2] The term subpopulation will refer to both its label d and the set of items in the subpopulation

w_i may be the flow's recorded bytes, or its packets, or a unit weight, depending on the estimation accuracy goals.

2.1 Fair Allocation

Following terminology common in resource sharing, we consider the subpopulation sizes $N = \{n_d \geq 0 : d \in \mathcal{D}\}$ as a set of *demands* over which the budget k is to be partitioned according to the allocations $K = \{k_d \geq 0 : d \in \mathcal{D}\}$.

A feasible allocation K^0 is said to be *Max-Min Fair* (MMF) [18] if for any other feasible allocation K, then for all SPs d such that $k_d > k^0_d$, there exists some SP d' such that $k_{d'} < k^0_{d'} \leq k^0_d$. In other words, allocating more than the MMF allocation to any SP, can only be done at the expense of reducing the allocation to some smaller SP.

If the allocations were not constrained to be integers, the well known progressive filling algorithm could be used to construct the MMF allocation [2]. Starting with all $k_d = 0$, all allocations are increased at the same rate until one or more of the SPs d are satisfied ($k_d = n_d$). Then the procedure repeats, dividing the remaining budget amongst the unsatisfied SPs, and so on, until either all SPs are satisfied or the budget is used up. However, the MMF allocation is not generally integral. This is problematic in our motivating example since flow records are indivisible. A possible workaround would be to use an integral allocation close to the MMF allocation. However, when allocating to a dynamic stream of demands, we must deal systematically with each discrete arrival (each flow record) rather than accumulate unfairness through successive uncontrolled approximations. Therefore, we will work with a more general notion of fairness, namely, Lexicographic Max-Min Fairness (see [3, 32]), that naturally accommodates discreteness.

Let $T(K)$ denote the elements of an allocation K sorted in non-decreasing order. K is said to be *lexicographically greater than* K', written $K \succ K'$, if the first non-zero component of $T(K) - T(K)'$ is positive. We say K is *lexicographically equal to* K' if $T(K) = T(K')$. We write $K \succeq K'$, if either $K \succ K'$ or $T(K) = T(K')$. With these definitions, the feasible allocation K^0 is *Lexicographically Max-Min Fair* (LMMF) if $K^0 \succeq K$ for any other feasible allocation K. LMMF enables handling the case where integer constraints prevent the MMF solution from existing, but several feasible allocations are close to it. As a simple example, consider a single unit budget that must accommodate two unit demands. The MMF allocation $K = (1/2, 1/2)$ is not allowed by the integer constraints, but the allocations $(1, 0)$ and $(0, 1)$ are both LMMF.

2.2 Skewness, Sampling and Estimation

Because of the observed heavy-tailed distribution of flow bytes, several streaming weighted sampling algorithms currently are used for flow record sampling, using byte size as the sampling weight. Threshold Sampling [14] performs independent weighted sampling. The related Priority Sampling [16] samples into a fixed size reservoir, and VarOpt [10]—described more completely below—does so with optimal estimation variance that is as good as any offline sampling scheme, and with computational cost smaller than previous related methods [6, 38]. All these methods can provide uniform sampling as a special case using unit weights.

All the schemes mentioned provide unbiased estimates of the sums of weights over any subset of items using the Horwitz-Thompson approach [21]. Let item i of weight w_i be sampled with marginal probability p_i. Each sampled w_i is replaced with its unbiased estimate (or adjusted weight) $\widehat{w}_i = w_i/p_i$; non-sampled items have adjusted weight 0. A generic query is to find the total weight of a subset S of items that satisfy some predicate, e.g., the total bytes of flows that use a specified protocol, The weight sum $W(S) = \sum_{i \in S} w_i$, has an unbiased estimator

$$\widehat{W}(S) = \sum_{i \in S} \widehat{w}_i = \sum_{i \in \widehat{S}} w_i/p_i \qquad (1)$$

where $\widehat{S}$ is the intersection of S with the random sample. This is a powerful construction because S need not be known at the time of sampling, enabling retrospective analysis.

2.3 Variance Optimal Sampling

Our methods use sampling to control the reservoir occupancy. Once the reservoir contains k items, any new item is provisionally added, then one random item is removed to return the occupancy to k. The weights of the surviving items are adjusted as above. This paper uses the VarOpt sampling primitive [10]. Its action VarOpt_k on a set Ω of $k+1$ weights w_i is recorded as Algorithm 1 below. Sampling is IPPS (Inclusion Probability Proportional to Size), item i being sampled with probability $p_i = \min\{1, w_i/\tau\}$, where τ is the unique value for which $\sum_i p_i = k$, i.e., k items are selected. Thus all large items (those i with $w_i \geq \tau$) are sampled with probability 1, while the remaining small items are sampled with probability proportional to size. This embodies the requirement from Section 2.2 to sample larger items. (Unit weights are used when uniform sampling is required). Sampling proceeds by picking one item i for discard with probability $1-p_i$. This is achieved by dividing up the unit interval into subintervals of length $1-p_i$, (note $\sum_i (1-p_i) = 1$), and discarding the item whose interval contains a random point chosen uniformly. Each of the k remaining weights are adjusted to their new value $w_i/p_i = \max\{w_i, \tau\}$. Note that VarOpt works with the item weights presented to it. *It does not use or require any assumptions concerning the statistical distribution of item weights.*

Used recursively, VarOpt samples exactly k out of any $n > k$ items. In addition to the estimate for individual weights being unbiased, the estimate of the total is exact and VarOpt minimizes the average estimation variance over weight subsets of a given size with respect to *any* unbiased estimator, including even offline algorithms. Furthermore, it has efficient implementation; its computational cost is no worse that $O(\log k)$ for any single new item, and in fact $O(\log \log k)$ when amortized over k items.

Algorithm 1: $\text{VarOpt}_k(\Omega)$ where $|\Omega| = k + 1$.

```
find $\tau$ be such that $\sum_{i=1}^{k+1} \min\{1, w_i/\tau\} = k$
for $i = 1, \ldots, k+1$ do $p_i \leftarrow \min\{1, w_i/\tau\}$,
$w_i \leftarrow \max\{\tau, w_i\}$
generate uniformly random $r \in U(0,1)$
maximize $j$ such that $\sum_{1 \leq i < j}(1 - p_i) \leq r$.
delete element $w_j$ from $\Omega$
```

2.4 Combined Estimation

In current network management practice, multiple sampling schemes are applied to flow records in support of different measurement applications; see e.g., [28]. Our proposed

Fair Sampling is not expected to supplant Undifferentiated Sampling; rather, it extends the applicability of measurement applications. Undifferentiated Sampling will remain important for applications that must focus on the largest traffic subpopulations. Thus it is natural to ask: *If both fair and undifferentiated sampling are performed, can the samples be combined to form an estimator that is better than either individually?* We will investigate this question experimentally in Section 6 using a combined estimator, proposed in general from in [15], that we now describe.

Given different unbiased estimates $\{X_i\}$ of some X, with associated estimation variances $\{V_i\}$, any convex combination $\sum_i \lambda_i X_i$, $\lambda_i \geq 0$, $\sum_i \lambda_i = 1$, is also an unbiased estimate of X. Choosing $\lambda_i = 1/V_i/(\sum_j 1/V_j)$ minimizes estimation variance. But naively substituting estimated variances leads to pathologies. For VarOpt, an unbiased estimate of the variance in estimating a weight w is

$$v = w \max\{0, \tau - w\} \tag{2}$$

where τ is the sampling threshold. But large items (of weight $w \geq \tau$) have zero associated variance, while some small items may be missed, yielding a small or zero estimated variance which causes the associated estimate X_i to dominate.

Regularized variance estimators proposed in [15] avoid these pathologies. The *bounded variance estimator* exploits that, from (2), $\mathsf{Var}(X_i)$ bounded above by $\tau_i X$, where τ_i is the sampling threshold in force for X_i. This bound is tight in the limit of small weights. Approximating V_i by $\tau_i X$ results in the convex combination estimate

$$X^{(\mathrm{reg})} = \sum_i X_i \tau_i^{-1} / \sum_i \tau_i^{-1} \tag{3}$$

This extends to uniform sampling with probability p, using an effective τ of $1/p$ times the maximum observed weight.

3. FAIR SAMPLING AND ACCURACY

3.1 Estimation Accuracy Goals

Our first contribution is to determine the relationship between the desired type of accuracy goal for estimation and the type of method used to share the sampling budget across SPs. We show that the choice between goals of controlling relative vs. absolute error in estimation leads naturally to a corresponding choice between Max-Min Fair vs. Proportional Fair sharing of the sampling budget. Furthermore, Proportional Fair Sharing offers little or no advantage over undifferentiated sampling. We first establish the result for uniform independent sampling (i.e., unit weights), then show how it extends to weighted VarOpt sampling. To be specific, we express our motivation in terms of estimation from subpopulations of flow records.

(i) *What to estimate:* The weight sum of subsets of flows that constitute at least some fraction $\beta < 1$ of the flows in a SP. Motivation: Analysis thresholds for both short-term management (e.g., anomaly detection) and long-term management (e.g., capacity planning) are both commonly expressed in terms of relative usage or changes thereof.

(ii) *Estimation accuracy:* two alternative measures of error estimating a fraction β of traffic: (ii$^{[abs]}$) the mean square *absolute* error; or (ii$^{[rel]}$) the mean square *relative* error. The importance of relative error when estimating over SPs of different sizes has been emphasized in [26].

(iii) *Balance across* SP*s:* to balance goals (i) and (ii) by minimizing the average error across SPs.

3.2 Fairness/Accuracy in Uniform Sampling

Consider an SP d of n_d items from which k_d on average are sampled uniformly and independently. Suppose we wish to estimate the size m_d of a subset of items satisfying some predicate. Using Horwitz-Thompson [21], an unbiased estimate for m_d is $\widehat{m}_d n_d / k_d$, where $\widehat{m}_d$ is the number of the m_d items that are sampled. The estimation variance is

$$V(n_d, k_d, m_d) = m_d \left(\frac{n_d}{k_d} - 1 \right) \tag{4}$$

Note the variance for undifferentiated sampling of the aggregate of n flows with sampling budget k is $V(n, k, m_d)$.

Theorem 1. *Consider the problem of estimating the size $m_d = \beta n_d$ of a fraction β of items in each* SP *d.*

(i) The average across all SP*s d of the absolute estimation variances $V(n_d, k_d, \beta n_d)$ is minimized by $k_d^{[abs]} = k n_d / n$. With this allocation, the ratio of the corresponding variance to that for undifferentiated sampling from* SP *d is*

$$\frac{V(n_d, k_d^{[abs]}, m_d)}{V(n, k, m_d)} = 1. \tag{5}$$

(ii) The average across all SP*s d of the relative estimation variances $V(n_d, k_d, \beta n_d)/(\beta n_d)^2$ is minimized by the Max-Min Fair Allocation $k_d^{[rel]}$ of the budget k over the* SP*s $\{n_d\}$. The ratio of the corresponding variance to that for undifferentiated sampling from* SP *d is*

$$\frac{V(n_d, k_d^{[rel]}, m_d)}{V(n, k, m_d)} \leq \frac{\max\{0, |\mathcal{D}| n_d - k\}}{n - k}, \tag{6}$$

with $V(n_d, k_d^{[rel]}, m_d) = 0$ if d is fully satisfied.

We draw two conclusions from Theorem 1. Firstly, from (5), proportional sharing does no better in terms of accuracy than undifferentiated sampling. Secondly, (6) tells us that for SPs contributing smaller than average offered load ($n_d < n/|\mathcal{D}|$), allocating equal sampling budget to each SP yields lower variance than undifferentiated sampling, while for SPs carrying larger than average load ($n_d > n/|\mathcal{D}|$) the reverse may be true.

3.3 Fairness/Accuracy in Weighted Sampling

Although Theorem 1 was proved for independent sampling with uniform weights, its conclusions extend to size-weighted sampling with a fixed budget. It is shown in [36] that in weighted VarOpt sampling of k_d from n_d items, the estimation variance of the weight of $m_d \leq n_d$, averaged over all subsets of m_d items, is bounded above by

$$U(n_d, k_d, n_d, X_d) = \frac{m_d (n_d - m_d) X_d^2}{n_d (n_d - 1) k_d} \tag{7}$$

where X_d is the total weight of the subpopulation. This bound is asymptotically tight when each item weight is $o(X_d/k)$. Assuming a common distribution of weights over items, take X_d proportional to n_d, and find $U(n_d, k_d, n_d, X_d) = a(\beta) V(n_d, k_d, m_d) + b$ where $a(\beta)$ is constant up to a term $O(1/n_d)$ and b is independent of k_d. Thus the analysis of Theorem 1 holds in the same approximation.

4. STREAMING FAIR SAMPLING

Section 3 has shown how fair allocation of the sampling budget uniformizes relative estimation variance over different subpopulations. The choice of sampling method is left open. For example, each subpopulation could be sampled offline using VarOpt to select a sample of size equal to its fair allocation. We now realize Fair Sampling as a reservoir sampling algorithm when the population is presented as a stream, and hence the demands (subpopulation sizes) are not known in advance. Section 4.1 describes a streaming fair allocation algorithm that adaptively allocates the sampling budget, while Section 4.2 uses sampling as a discard mechanism when the allocation to a SP is reduced.

4.1 Streaming Fair Allocation

Adapting our previous notation, we now assume that items arrive in a stream labeled by $i = 1, 2, \ldots, n$, with item i belonging to subpopulation $d^i \in \mathcal{D}$. From the stream we wish to select k items in total, with each SP d obtaining its fair allocation k_d of the total slots k. The allocations are computed progressively as items arrive, so that, after k initial items have arrived, the allocations at any time will be exactly consumed by the sampled items; only a single pass through the data is performed. Our Algorithm 2 can be summarized as follows: *Allocate budget to* SP*s until exhausted; thereafter accommodate new* SP*s by decrementing the allocation of a* SP *with maximal current allocation.*

Algorithm 2: Stream Fair Allocation of a Budget of Size k

```
$D \leftarrow \emptyset$;
while new item in SP $d$ do
    if $d \notin D$ then
        include $d$ in $D$
        $k_d \leftarrow 0$
    $k_d \leftarrow k_d + 1$
    if $\sum_{d \in D} k_d > k$ then
        select $d_* \in D$ that maximizes $k_{d_*}$
        $k_{d_*} \leftarrow k_{d_*} - 1$
```

$K = \{k_d : d \in \mathcal{D}\}$ will denote the generic allocation. Let $\mathcal{X}(N, k)$ denote the feasible allocations of the budget k to demands $N = \{n_d : d \in \mathcal{D}\}$ of the subpopulations, and let $\mathcal{X}_d(N, k) \subset \mathcal{X}(N, k)$ denote those feasible allocations in which the demand n_d is fully satisfied, i.e., $k_d = n_d$. Let $\mathbf{1}_d$ denote the unit demand from SP d. N^i is the cumulative demand due to the first i arrivals, i.e., $n^i_d = \#\{j \leq i : d^j = d\}$. Let $K^i = \{k^i_d : d \in \mathcal{D}\}$ denote any allocation provided by Algorithm 2 from the first i items. Notationally, it will be convenient for $\mathcal{D}$ to be the fixed set of all possible demands, so $k^i_d = 0$ for demands d not represented in the first i items. Of course, actual implementations need only retain labels for demands currently represented in the reservoir.

Theorem 2. *For each i, K^i is LMMF for the demands $N^i = \{n^i_d : d \in \mathcal{D}\}$.*

In Algorithm 2, K^{i+1} is generated from K^i by composing two steps. Firstly, the arriving increment to demand n_d is added, then the allocation to one of the largest current demand n_{d_*} is decremented. We write these steps as

$$K \mapsto K + \mathbf{1}_{d^{i+1}} \mapsto K + \mathbf{1}_{d^{i+1}} - \mathbf{1}_{d_*} \tag{8}$$

The proof of Theorem 2 proceeds by demonstrating that each of these two steps, subject to the conditions of our problem, maintains lexicographic order, as we now state. We will need the notation $\Sigma K = \sum_{d \in \mathcal{D}} k_d$. The proofs of all the theorems are deferred to Section 10.

Theorem 3. *Suppose $K \succeq K'$ and $k_d \leq k'_d$. Then $K + \mathbf{1}_d \succeq K' + \mathbf{1}_d$.*

Theorem 4. *Let $\Sigma K \geq \Sigma K'$, $K \succeq K'$ and d_* maximize k_d. Then $K - \mathbf{1}_{d_*} \succeq K' - \mathbf{1}_d$ for all $d \in \mathcal{D}$.*

4.2 Stream Sampling and Budget Deallocation

Algorithm 2 extends to *Stream Fair Sampling* by maintaining a reservoir of size k of selected items. Each arriving item is added to the reservoir; when an allocation k_{d_*} is decremented, an item from subpopulation d_* is deleted from the reservoir. Although any method that deletes a single item could be used, sampling is a natural choice because the statistical consequences of deletion for estimation can be controlled. We use VarOpt sampling because it can delete exactly one item, can be efficiently implemented, and has minimal average variance for estimation of subset sums of a given size. For estimating bytes from flow records, the reported bytes should be used as a sampling weight. Other applications may be better served by a different assignment of weight, e..g. uniform weights for estimating flow counts.

We incorporate VarOpt as the discard step for Algorithm 2 as follows. We represent each item in the stream as a pair (d, w) where d is the subpopulation the item belongs to, and w is the weight. D is the set of subpopulations observed and Ω_d is the multiset of weights selected from subpopulation d. The sizes $|\Omega_d|$ in Algorithm 3 play the same role as the allocations k_d in Algorithm 2. Here we store only the weights w; obviously, one could store the (d, w) plus any other information carried by the item.

Algorithm 3: Stream Fair Sampling into a Reservoir of Size k

```
$D \leftarrow \emptyset$;
while new item $(d, w)$ arrives do
    if $d \notin D$ then
        include $d$ in $D$
        $\Omega_d \leftarrow \emptyset$
    include $w$ in $\Omega_d$
    if $\sum_{d \in D} |\Omega_d| > k$ then
        select $d_* \in D$ that maximizes $|\Omega_{d_*}|$
        $\textsc{VarOpt}_{|\Omega_{d_*}|-1}(\Omega_{d_*})$
        if $|\Omega_{d_*}| = 0$ then
            remove $d_*$ from $D$
```

Online and offline fair sampling are equivalent in a statistical sense we now discuss. Concerning solely the sampling aspects, it was shown in [10] that progressively VarOpt sampling a stream of n items into a reservoir of capacity k is statistically equivalent to single VarOpt sampling of k out of n items. It should now be clear the Fair Allocation Algorithm 2 enjoys a similar recurrence property; after n items have been processed, the final allocations K^n are LMMF for the demands N. From this it is largely a matter of formalism to show that the Stream Fair Sampling Algorithm 3 satisfies an analogous property. The final sample is equivalent to

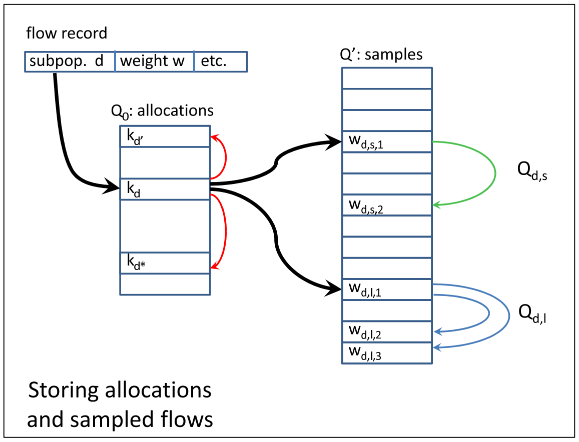

Figure 1: Data Structures for allocations k_d (in Q_0) and samples (in Q'). Thick arrows map from flow record demand d to allocation k_d in Q_0 and onward to large and small item sets $Q_{d,l}, Q_{d,s} \subset Q'$. Parent to child mappings within BSTs shown as thin arrows.

one obtained by (i) finding LMMF allocations $\{k_d\}$ for the demands $\{n_d\}$, then (ii) making a VarOpt selection of k_d out of n_d items for each subpopulation d. The equivalence is that the distribution of sampled items is the same when conditioned on obtaining the same LMMF allocations.

5. IMPLEMENTATION AND COSTS

This section describes data structures that can support Stream Fair Sampling Algorithm 3, and their associated computational costs per item. Although the collector and mediator environments are less resource stringent than on a router, in view of the high arrival rate of flow records, it is important to demonstrate that Stream Fair Sampling does not add per item time complexity above that of existing Undifferentiated Stream Sampling algorithms.

We first review the maintenance of a single VarOpt$_k$ sample set from [10]. $k+1$ storage locations are required, i.e., k for the items stored, plus 1 for the new item. Each location stores an item weight, associated properties, and address pointers needed to maintain the data structure. Summarizing from [10], the data structure comprises two Binary Search Trees (BSTs) of total size $k+1$: Q_l which stores large items (with weight greater than the current threshold τ) and Q_s for the remaining small items. The VarOpt$_k$ sampling operation involves refreshing the τ value, selecting an item for deletion, and transferring any items that change order with respect to the new threshold, between Q_l and Q_s, as appropriate. These operations cost at most $O(\log k)$ per arriving item, amortizing to $O(\log\log k)$ over k items.

For Fair Sampling, we require a data structure Q_0 that maintains the current allocations $\{k_d : d \in \mathcal{D}\}$; see Figure 1. Q_0 is organized as a k_d-descending order priority queue Q_0 of size $|\mathcal{D}|$, implemented, e.g., as a self-balancing BST. The computation time for budget allocation is no worse than $O(\log|\mathcal{D}|)$ per arriving item. The subpopulation membership represented in each arriving stream item must be mapped to the associated location in Q_0. For a fixed set $\mathcal{D}$ of known subpopulations (e.g., interface identifiers), this mapping can be generated at initialization. Otherwise, a collision free hash of the subpopulation label d into Q_0 is used.

Finally, we describe the storage of the samples themselves in a data structure Q' which maintains VarOpt samples for all SPs in $k+1$ locations. The entry in Q_0 for SP d includes its current sampling threshold τ_d and two pointers to the roots of two BSTs, $Q_{d,l}$ and $Q_{d,s}$, within Q' that maintain the large and small items for SP d. A pointer u to the unused location in Q' also is maintained. The operation of Algorithm 3 is then the following:

- ○ *Insertion:* An arriving item with weight w from SP d is mapped to an the appropriate location in Q_0 based on its SP label d, where the item count is incremented. The item is then inserted the empty location in Q', and incorporated in either $Q_{d,l}$ or $Q_{d,s}$, depending on the order of its weight w relative to τ_d.
- ○ *Allocation:* When storage is full, the SP d_* whose allocation is to be decremented is the next item in the priority queue of Q_0. The allocation k_{d_*} then is decremented.
- ○ *Sampling:* One item from SP d_* is removed by operation of VarOpt using the priority queues $Q_{d_*,l}$ and $Q_{d_*,s}$. The threshold τ_{d_*} is updated.
- ○ *Cleanup:* The unused-location pointer u is updated. If $k_{d_*} = 0$, the SP label d_* can be removed from Q_0.

Since at most k items are stored from each SP, the operation of Algorithm 3 on Q' has the same cost as VarOpt$_k$.

To summarize: The allocation cost is $O(\log|\mathcal{D}|)$, which is no worse than $O(\log k)$ even when then set of SPs $\mathcal{D}$ is not known in advance. The sampling cost is no worse than $O(\log k)$ per item. In a setting where no SP demand is fully satisfied, each has about $k/|\mathcal{D}|$ samples, so the sampling cost is in fact no worse than $O(\log k/|\mathcal{D}|)$.

6. EVALUATION: ACCESS ROUTER FLOWS

Access routers are situated at the edge of the ISP network where they provide customer access to the network, in some cases supporting hundreds of interfaces. Interface speeds can be diverse, ranging from, e.g., Gigabit Ethernet (10^9 bps) down to T1 (1.5×10^6 bps), i.e., roughly 3 orders of magnitude. The aim of this section is to illustrate the relative advantages of Fair Sampling and Undifferentiated Sampling in serving a generic class of traffic queries over the different rate interfaces. Actually, this is already determined by the performance analysis of Theorem 5 in Section 7 following. That result can be used to practically dimension measurement systems, relating traffic rates, sampling budget, and accuracy goals. Nevertheless, we will confirm its prediction on our data; see Figure 4 in Section 7.

6.1 Interface Flow Data & Heterogeneity

The evaluation used passive flow measurements on traffic traversing an access router of a major ISP. Each flow record summarized a flow that entered the router through one of 240 active customer-facing interfaces and egressed on a single uplink into the network core. The flow records, approximately 28.4 Million in all, were compiled without packet sampling by a standalone device observing the uplink during a 24-hour period in 2011. Since the flow records were not compiled at the access router, they did not report which router interfaces were traversed. Instead, this was inferred during preprocessing by longest prefix matching against the

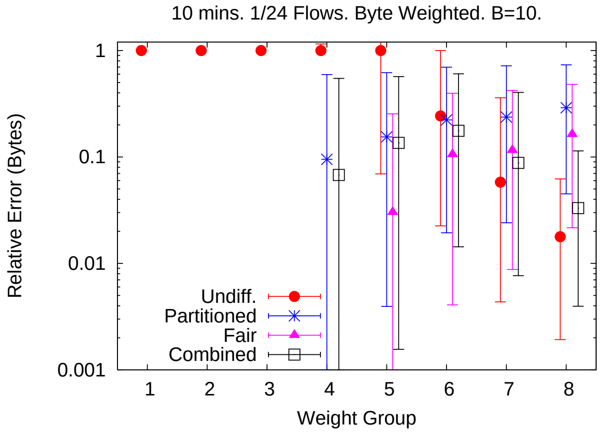

Figure 2: Byte Relative Estimation Error, Byte Weighting and average 1 in 24 flow sampling, for US, PS, FS and CS. 10 minute granularity, spatial granularity $B = 10$. Median, 10%-ile, 90-%ile.

forwarding table downloaded periodically from the access router. This preprocessing stage created a working set of derived flow records for subsequent use, that contained only anonymized versions of the IP addresses and interfaces.

The router interface line speed varied over more that three orders of magnitude. Utilization varied considerably over time and interfaces, with (non-zero) hourly utilizations varying over 5 orders of magnitude. Furthermore, large variations occurred within subsets of equal-speed interfaces. Hence, line speed is not an effective proxy for actual traffic rates, instead motivating the distribution of budget according to actual traffic demands.

This data does not represent all traffic on each access router interface, since traffic that passes directly between interfaces on the access router was not measured. However, the 5 minute average rate of interface traffic derived from SNMP polling of the access router exhibited similar variability. Furthermore, substituting forwarding tables from adjacent time periods during preprocessing scarcely changed the inferred traffic distribution, so potential staleness of the interface mapping was not an issue.

6.2 Comparisons and Reference Methods

Our principal comparison of *Fair Sampling* (FS) is with the state-of-the-art stream sampling method for completed flow records, namely *Undifferentiated Sampling* (US), in which VarOpt sampling is applied to the whole population of flow records oblivious of SP membership[3]. Two other variants are also considered. Firstly, in *Partitioned Sampling* (PS), each interface is assigned a fixed-size sampling budget in a private reservoir. The purpose of this comparison is to examine the benefits of differentially allocating the sampling budget amongst interfaces in isolation from the benefits of redistributing unused budget between interfaces. Such an approach might conceivably be used if flow records were sampled separately at each interface. Secondly, the likely coexistence of FS and US in deployments motivates using *Combined Sampling* (CS)—see Section 2.4—to combine the two sample sets obtained from FS and US. This shows the merits of using both FS and US as opposed to using only one method but with increased budget.

6.3 Subset Sum Queries and Accuracy

Computing weight sums over subsets of flow records forms the basis of many complex database queries. One cost of sampling is reduced accuracy in estimating smaller sums. Following the accuracy goals stated in Section 3, we frame our evaluation in terms of the ability to estimate the weight of subsets of flows representing a fraction β of flows on each interface. Multiple such subsets were formed by assigning each occurring anonymized IP local address at random to one of B bins associated with the interface traversed, i.e., with $B = 1/\beta$. Let $S_{d,b}$ denote the subset of flows assigned to bin $b \in \{1, \dots, B\}$ at interface d. Let $W_{d,b} = W(S_{d,b})$ denote the corresponding subset sum of flow weights, and $\widehat{W}_{d,b}$ its unbiased estimator. For occupied bins, the associated accuracy is the relative error $\rho_{d,b} = |1 - \widehat{W}_{d,b}/W_{d,b}|$.

Rather than focus on particular measurement applications, we abstract a detection problem: How reliably can it be detected that a weight sum $W_{d',b'}$ (within a SP d' of aggregate weight $W_{d'}$) exceeds a weight sum $W_{d,b}$ (within a SP d of aggregate weight W_d) by a factor $r > 1$, i.e., $W_{d',b'} = rW_{d,b}$. We say detection fails when $\widehat{W}_{d',b'} \le \widehat{W}_{d,b}$.

We now compute some target detection rates for evaluation, using two bounds. Firstly, variance bounds from Section 1.5. of [10] yield $\mathsf{Var}\, \widehat{W}_{d,b} \le W_d \cdot W_{d,b}/k_d$, and similarly for (d', b'). Secondly, Chebyshev's inequality states $\Pr[|Y - \mathsf{E}[Y]| \ge a] \le \mathsf{Var}(Y)/a^2$ for a random variable Y and $a > 0$. Combining these, the tail distribution of the relative errors is bounded as $\Pr[\rho_{d,b} \ge \varepsilon] \le q := W_d/(W_{d,b}k_d\varepsilon^2)$, while the probability of detection failure is bounded as

$$\begin{aligned}\frac{\mathsf{Var}\, W_{d,b} + \mathsf{Var}\, W_{d',b'}}{(W_{d,b} - W_{d',b'})^2} &\le \frac{\frac{W_d}{k_d W_{d,b}} W_{d,b}^2 + \frac{W_{d'}}{k_{d'} W_{d',b'}} W_{d',b'}^2}{(W_{d,b} - W_{d',b'})^2} \\ &= p(q, \varepsilon, r) := q\varepsilon^2(r^2+1)/(r-1)^2\end{aligned}$$

Treating for simplicity the tail bound on $\rho_{d,b}$ as an approximation, then requiring at most a fraction q of the relative errors to exceed ε for all d, results in detection failure with probability roughly $p(q, \varepsilon, r)$. As illustrative values, we take $q = 0.1$ and $\varepsilon = 0.5$ (the 90%-ile of the relative error is to be 0.5 or less), yielding a detection failure rate of 6% for $r = 3$.

6.4 Evaluation Parameters

- *Sample Weighting:* Three variants of flow weight for VarOpt were used: byte, packet, and uniform.
- *Temporal Granularity:* 24 hours data was divided into measurement periods using either 144 windows of 10 minutes, or 24 windows of 1 hour.
- *Subset Query Granularity:* We report experiments with $B = 10$; we found the same qualitative relations amongst the performance of the sampling methods at $B = 100$, although accuracy was generally poorer.
- *Reservoir Size:* We use a constant reservoir size equivalent to a daily average flow sampling rate of 1 in 24 flows. This is

[3] We restrict comparison to US since it is the method of *flow record* sampling on which we wish to improve in practice. Related methods described in Section 8 instead concern packet sampling, or do not allow direct expression of balanced accuracy goals over SPs, or address skewness in flow size rather than in subpopulation size.

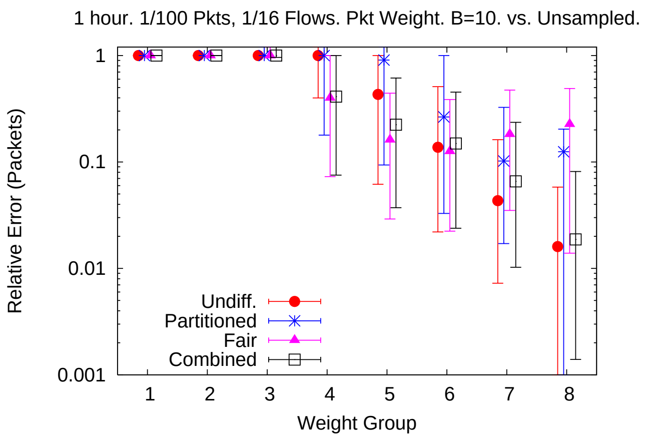

Figure 3: Packet Sampling. Relative Estimation Error for US, PS, FS, and CS. 1/100 packet sampling, 1/16 flow sampling. Packet weighting. Temporal Granularity 1 hour; Spatial: $B = 10$. Median, 10%-ile, 90-%ile.

in line with operational values and corresponds to significant reduction in both storage required and execution time for flow selection operations in database queries.

6.5 Evaluation Results

The evaluation results are presented as follows: In each parameter configuration, the experiments generated a collection of relative error statistics $\rho_{d,b}$ over all interfaces d. To break out the dependence of accuracy on subpopulation size, the errors $\rho_{d,b}$ are grouped according to true weight on their interfaces represented by the group index $g(d) = \lfloor \log_{10} W_d \rfloor$. We plot the median, 10%-ile, and 90%-ile of the $\rho_{d,b}$ values within each such weight group. In the resulting *Rangeplot* the absolute sizes are obfuscated through the adjustment $g(d) \mapsto 1 + g(d) - \min_{d'} g(d')$. A further comparison measure between sampling methods is the *Improvement Ratio*, the fraction of relative errors $\rho_{d,b}$ that are reduced, by a given sampling method, compared to another.

Figure 2 shows rangeplots of $\rho_{d,b}$ for US, PS, FS and CS under the same total budget. We see immediately:

(i) US is not able to furnish any useful estimates for medium and smaller weight groups (group 5 and lower).

(ii) FS yields negligible errors ($< 10^{-3}$) for small weight groups (2 and below) with error climbing greater than US for the largest groups, but still within target (i.e., 90%-ile relative error less than 0.5).

(iii) PS is uniformly worse than FS with error roughly twice as large for weight groups 4 and above.

(iv) CS provides a good compromise between FS and US, meeting or nearly meeting accuracy goals in all weight groups.

In terms of improvement ratios, FS improved on US in 84% of $\{d, b\}$, while the reverse held true in only 15% of cases. CS provided another 5% improvement against US mainly for a relatively small number of high weight estimates.

Space limitations preclude graphical results on two further parameter configurations. Firstly, the same general relative performance between the methods was found for uniformly weighted (as opposed to byte weighted) sampling. Secondly, the performance of CS is relatively insensitive to small to medium changes in the allocation of total budget between FS and US because (3) de-emphasizes estimates with higher thresholds τ corresponding to smaller sampling budgets.

7. ANALYSIS & PACKET SAMPLING

In our motivating example from network measurement, routers often compile flow records from a sampled packet stream, e.g., in Packet Sampled NetFlow [9]. For this reason, it is important to determine how much packet sampling offsets the advantages of downstream fair sampling of flow records for smaller SPs. To this end, we compute the relative estimation variance for a fraction β of the flows under the composition of 1 in M packet sampling followed by VarOpt flow record sampling. This extends some previous work from [13] which applied to the simpler case of threshold sampling.

Theorem 5. *Let S be a subset of flows comprising at least a fraction β of both bytes and packets in* SP *d, subject to 1 in M packet sampling, then* VarOpt *sampling of k_d flows with byte weighting. Assume N_d total packets from all original flows in the* SP*. Then the relative estimation variance of the total bytes in S is bounded above by:*

$$R_d^2(M) = \beta^{-1}\left(k_d^{-1} + (1+k_d^{-1})\frac{(1+r^2)(M-1)}{N_d}\right) \tag{9}$$

with r^2 the relative variance of the packet size distribution.

For system dimensioning, bounds on the relative variances of Undifferentiated (resp. Fair) Sampling are obtained by inserting $k_d^{[abs]}$ (resp. $k_d^{[rel]}$) from Theorem 1 into (9). In fact, for Fair Sampling, using $k/|\mathcal{D}|$ in place of $k_d^{[rel]}$ suffices.

We now establish the regime in which Fair Sampling is advantageous even after packet sampling. Recall from Theorem 1 the relative variance for flow record sampling alone is about $R_d^2(1) = 1/(\beta k_d)$. Assuming for simplicity that all demands can use their allocation, then Fair Sampling sets $k_d = k/|\mathcal{D}|$ while Undifferentiated Sampling has $k_d = kn_d/n$ on average, leading to lower relative variance for Fair Sampling when $n_d \leq n/|\mathcal{D}|$, or equivalently taking N_d proportional to n_d, when $N_d \leq N_{\text{tot}}/|\mathcal{D}|$, where $N_{\text{tot}} = \sum_d N_d$ is the total number of packets on all subpopulations.

But unless, due to packet sampling, the second term in (9) is smaller than the first for $k_d = k_d^{[rel]}$, then estimation variance of packet sampling will dominate that of flow sampling for all d where Fair Sampling has lower variance than Undifferentiated Sampling. Summarizing, SPs d for which

$$(1+r^2)Mk/|\mathcal{D}| < N_d < N_{\text{tot}}/|\mathcal{D}| \tag{10}$$

can expect to see a benefit from Fair Sampling. (Here we have simplified for $k, M \gg 1$). This requires the range of possible N_d be non-empty, i.e., $(1+r^2)k < N_{\text{tot}}/M$. To conclude: *For Fair Sampling to benefit after packet sampling, the aggregate number of packets sampled in packet sampling must exceed $(1+r^2)$ times the capacity for flows in the sampling reservoir.*

To confirm the impact of packet sampling on estimation accuracy, we repeated the experiments of Section 6 but simulating 1 in M=100 packet sampling prior to flow sampling at an average rate on 1 in 16. Since the flow data did not contain the byte sizes of individual packets, estimation focused

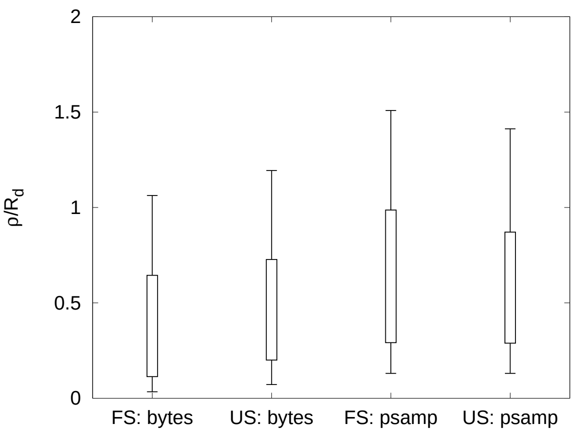

Figure 4: Ratios $\rho_{d,b}/R_d$ for FS, US in configurations of Figures 2 and 3 (w/ packet sampling). Boxplot with 25%-ile, 75%-iles, whiskers at 10%-ile and 90%-ile. Distribution near/below 1 confirms utility of (9).

instead on the number of packets in each subset $S_{d,b}$, using the number of sampled packets reported in a flow record as its sampling weight. Figure 3 compares US, PS, FS, and CS, with relative errors computed w.r.t. to the original stream before packet sampling. All methods perform badly (with relative error around 1 or greater) for small weights (groups 1–3), where only a few flows survive packet sampling. FS is more accurate than US for medium weight groups (4–6). Only CS and FS retain acceptable or near acceptable performance over weight groups 4 and higher. Estimation with FS improved on US in 46% of cases.

We now relate this behavior to the analysis of the useful regime for FS above. There were approximately N = 20 Million packets per hour, distributed over $|\mathcal{D}| = 240$ active interfaces, with $k = 10,000$ sampled flows per hour after 1 in $M = 100$ packet sampling. Thus from (10), using $r = 1$, we expect FS to improve accuracy for interface d if its hourly packet arrival rate N_d obeys: $8.4 \times 10^3 < N_d < 8.7 \times 10^5$. There were 80 such interfaces, comprising 1/3 of the total.

As a check on the predictive utility of (9) we computed the ratio $\rho_{d,b}/R_d$ over all sets $S_{d,b}$, using the actual fraction β rather than the targets $1/B$. The ratio should be distributed around or less than 1. This is confirmed in Figure 4 with boxplots of $\rho_{d,b}/R_d$ for FS and US in the configurations of Figure 2 and 3.

8. RELATED WORK

The general problem of fair resource allocation has attracted much attention in the networking context and beyond. There is extensive literature involving fairness in scheduling algorithms; see e.g., [2, 24, 30]. More recently, this approach has focused on developing areas of network management, including bandwidth allocation in wireless technologies, see e.g., [37], routing policy [31, 25], and ramifications of the choice of fairness criteria bandwidth sharing and congestion in the internet; see e.g., [23]. Some recent work that examines fairness properties of packet discard mechanisms seem somewhat suggestive of our approach [22, 27]; likewise for a recent approach to resource allocation in sensor networks [8]. However, these appear not to be directly applicable to our problem domain.

A number of papers have dealt with the differential allocation of packet sampling resources over individual flows. A starting point for these is Sample and Hold [17] in which packets are uniformly sampled, but a packet with a current cached key always is selected. Sketch Guided Sampling [26] and more recent implementations [40] counteract the inherent tendency of Sample and Hold to preferentially select longer flows, by decreasing the probability that a packet is sampled according to the currently measured flow length. The probability is chosen to uniformize the relative estimation error of different length flows. Our work is reminiscent of this general idea but different: We seek to minimize the average relative estimation error from general purpose samples of flow records of drawn from SPs of different volumes.

FlexSample [33] is a system for allocating a per-packet sampling budget over traffic SPs, including low-volume SPs, that match criteria set via a configuration language. Although the high-level motivation similar to ours, the detailed aims and context are different and lead to different approaches, as we now explain. Targeting the tight storage constraints of the router context, FlexSampling is bufferless—both packet selection and discard are final—with SP sampling probabilities dynamically adjusted according to smoothed past traffic rates. Unlike Fair Sampling, FlexSampling does not express quantitative accuracy goals for SP estimation; rather, these would depend on the SP sizes and whatever sampling probabilities the user specifies. Fair Sampling aims to yield general purpose SP flow summaries in the more resource relaxed environment typical of mediators and collectors. It does not sample packets. Working with a reservoir of provisional samples, Fair Sampling probabilities are determined by actual arrivals as a function of estimation accuracy goals. Being rate adaptive, FlexSampling distributes it budget allocation non-uniformly over the packets within a given SP, an effect which will be more pronounced after bursts of arrivals within the smoothing timescale. In contrast, Fair Sampling is uniform per unit of flow weight within each SP. Finally, the implementation of Fair Sampling is relatively simple compared with FlexSampling because it does not estimate SP rates.

Three other recent papers have dealt with broader issues of distributing sampling budget for network measurement over multiple locations. [35] and [5] considered the problem of setting rates for independent sampling in order to maximize a total utility of packets being sampled at least once. cSamp [34] uses a hash-based selection on flow keys to implicitly coordinate sampling over multiple locations in order to both partition resources over SPs and avoid duplicate measurement at different observation points.

9. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed Fair Sampling as a new approach to sampling flow records derived from rate heterogeneous subpopulations of traffic, e.g., across different interfaces of an access router, or different applications. Max-Min Fair Allocation of the sampling budget was shown to be optimal for minimizing average relative estimation error over SPs of flow records generated by the different traffic components. Fair Sampling is simple, flexible and performs better than a fixed partitioning of the reservoir. Compared with undifferentiated sampling, it greatly improves estimation accuracy for all but the largest traffic components while being no more computationally complex. A combined estimator reduced relative error in 89% of cases.

General collection and analysis infrastructures can have a complex structure in which selection of an item can, in conjunction with other items, trigger downstream usage of limited resources for analysis or even further measurements. This motivates a methodological extension of fair sampling to a network of budget nodes shared by subpopulations, with items subject to resampling at multiple nodes.

10. PROOFS OF THEOREMS

PROOF OF THEOREM 1. (i) minimize $\sum_d V(n_d, k_d, \beta n_d)/|\mathcal{D}| = |\mathcal{D}|^{-1} \sum_d \beta(n_d^2/k_d - 1/n_d)$ under the constraints on the k_d. This is equivalent to

$$\begin{array}{rl} \min: & \sum_d n_d^2/k_d \\ \text{such that}: & 0 \le k_d \le n_d, \quad \sum_d k_d \le k \end{array} \tag{11}$$

Since $k \to 1/k$ is convex, an elementary convex optimization yields that k_d is proportional to n_d, and hence $k_d = kn_d/n$. The statement concerning the ratio of variance follows because $V(n_d, k_d^{[abs]}, m_d) = m_d(n_d/k_d^{[abs]} - 1) = m_d(n/k - 1) = V(n, k, m_d)$.

(ii) We minimize $|\mathcal{D}|^{-1} \sum_d V(n_d, k_d, \beta n_d)/(\beta n_d)^2 = (|\mathcal{D}|\beta)^{-1} \sum_d (1/k_d - 1/n_d)$ under the constraints on the k_d. This is equivalent to

$$\begin{array}{rl} \min: & \sum_d 1/k_d \\ \text{such that}: & 0 \le k_d \le n_d, \quad \sum_d k_d \le k \end{array} \tag{12}$$

In the absence of the constraint $k_d \le n_d$, an elementary convex optimization would yield the solution the k_d is independent of $|\mathcal{D}|$ and hence $k_d = k/|\mathcal{D}|$. But when $k/|\mathcal{D}| > n_{d'}$ for the d' of minimal $n_{d'}$, this would result in a sampling probability $p_{d'} > 1$, Instead we instead set $k_{d'} = n_{d'}$, the repeat the optimization after removing the "small" interface d' from the sum and depleting the sampling resources from k to $k - n_{d'}$. We repeat until no small interfaces remain. (This must happen since $n > k$). Finally observe that the procedure we have described is just the usual progressive filling algorithm to compute the Max-Min Fair allocation of total resources k over the demands $\{n_d\}$. For "small" interfaces j whose demand is fully satisfied, the estimation variance is zero (all flows are sampled). The remaining interfaces are allocated at least $k/|\mathcal{D}|$, hence the bound follows. □

PROOF OF THEOREM 5. Let Φ_d denote the flows on interface d, and $S \subset \Phi_d$ the subset of interest. w_i is the byte size of flow i, $\widehat{w}_i$ its estimate after packet sampling, and $\widehat{y}_i$ the estimate of $\widehat{w}_i$ after flow sampling. By Lemma 1 in [13]:

$$\mathsf{Var}(\widehat{y}_i) \le \mathsf{E}[\widehat{w}_i \widehat{\tau}] + \mathsf{Var}(\widehat{w}_i) \tag{13}$$

where $\widehat{\tau}$ is now a random threshold for VarOpt that depends on all $\{\widehat{w}_i : i \in \Phi_d\}$. Setting $\widehat{W}_d = \sum_{i \in \Phi_d} w_i$:

$$k_d = \sum_{i \in \Phi_d} \min\{1, \widehat{w}_i/\widehat{\tau}\} \le \widehat{W}_d/\widehat{\tau} \tag{14}$$

Thus $\mathsf{E}[\widehat{w}_i \widehat{\tau}] \le k_d^{-1} \mathsf{E}[\widehat{w}_i \widehat{W}_d] = k_d^{-1}(w_i W_d + \mathsf{Var}(\widehat{w}_i))$. Thus so far $\mathsf{Var}(\widehat{y}_i) \le k_d^{-1} w_i W_d + (1 + k_d^{-1}) \mathsf{Var}\, \widehat{w}_i$.

A standard argument (see [13]) yields $\mathsf{Var}(\widehat{w}_i) = (M - 1) \sum_j x_j^2$, summing over the sizes x_j of packets j in flow i. From the negative covariance property of VarOpt:

$$R_d^2 = \frac{\mathsf{Var}(\sum_{i \in S} \widehat{y}_i)}{W(S)^2} \le k_d^{-1} \frac{W_d}{W(S)} + (1 + k_d^{-1}) \frac{\sum_j^S x_j^2}{W(S)^2}$$

with $\sum_j^S$ the sum over all $N(S)$ packets from S. The result follows since $\sum_j^S x_j^2/W(S)^2 = N(S)^{-1}\overline{x^2}/(\overline{x})^2$. □

PROOF OF THEOREM 2. Trivially $K^i = N^i$ is LMMF for $i = 1, \ldots, k$ since it majorizes elementwise any feasible allocation of k. Henceforth we consider $i \ge k$ and proceed by induction. We first show that $K^{i+1} \succeq K^i$. Write the non-decreasing ordered elements of $T(K^i)$ as $k^i_{(1)}, ..., k^i_{(|\mathcal{D}|)}$. The *index* of an element k of K will refer to its position in the vector K. The rank $r(k)$ of the element k is the lowest index of all elements with the same value k. Let D^i_* denote the set of demands that maximize $d \mapsto k^i_d + (\mathbf{1}_{d^{i+1}})_d$. If $d^{i+1} \in D^i_*$ then $T(K^{i+1}) = T(K^i)$ and we are done.

Assume now that $d^{i+1} \notin D^i_*$. Write a for rank of $k^i_{(d^{i+1})}$ in $T(K^i)$, and b for the rank of $1 + k^i_{d^{i+1}}$ is $T(K^{i+1})$. Then $T(K^{i+1})_{(j)} = k^i_{(j+1)} \ge k^i_{(j)}$ for $j = a, \ldots b-1$ while $T(K^{i+1})_{(b)} = 1 + k_{(a)} \ge k^i_{(b)}$, all lower index elements being unchanged. Because $k_b^{i+1} = 1 + k^i_a$, the strict inequality $k^{i+1}_{a'} > k^i_{a'}$ must hold for some $a' \in \{a, \ldots, b\}$ and hence $K^{i+1} \succ K^i$. Now $\mathcal{X}(N^i, k) \subset \mathcal{X}(N^{i+1}, k)$ since $n^i_d \le n^{i+1}_d$ for all $d \in \mathcal{D}$. Hence for each $K \in \mathcal{X}(N^i, k)$ we have shown that $K^{i+1} \succ K^i \succeq K$.

It remains only to show that $K^{i+1} \succeq K'$ for all $K' \in \mathcal{X}(N^{i+1}, k) \setminus \mathcal{X}(N^i, k)$. Any such K' must be an element of $\mathcal{X}_{d^{i+1}}(N^{i+1}, k)$, since, if not, $k'_{d^{i+1}} \le n^{i+1}_{d^{i+1}} = n^i_{d^{i+1}} - 1$ which would mean $K' \in \mathcal{X}(N^i, k)$. Hence $K' = K + \mathbf{1}_{d^{i+1}} - \mathbf{1}_d$ for some $K \in \mathcal{X}_{d^{i+1}}(N^i, k)$ and $d \ne d^{i+1} \in \mathcal{D}$. Thus it suffices to show that:

(i) $K^i + \mathbf{1}_{d^{i+1}} \succeq K + \mathbf{1}_{d^{i+1}}$ for any $K \in \mathcal{X}_{d^{i+1}}(N^i, k)$. This follows from Theorem 3 since we assume $K^i \succeq K$ for any $K \in \mathcal{X}(N^i, k)$.

(ii) $K^i + \mathbf{1}_{d^{i+1}} - \mathbf{1}_{d_*} \succeq K + \mathbf{1}_{d^{i+1}} - \mathbf{1}_d$ for any $K \in \mathcal{X}_{d^{i+1}}(N^i, k)$, $d_* \in D^i_*$ and $d \in \mathcal{D}$. Given (i), this follows from Theorem 4

□

PROOF OF THEOREM 3. Let $a = \min\{i : k_{(i)} > k'_{(i)}\}$, so that $k_{(j)} = k'_{(j)}$ for $j < i$. **Case: $\boldsymbol{k_{(a)} < k_d}$.** Since $k_{(a)} < k_d \le k_{d'}$, then $T(K + \mathbf{1}_d)_{(i)} = k_{(i)}$ and $T(K' + \mathbf{1}_d)_{(i)} = k'$ for $i \le a$. Hence $K + \mathbf{1}_d \succeq K' + \mathbf{1}_d$.

Case: $\boldsymbol{k_{(a)} = k_d}$. Since $k'_d \ge k_d = k_a > k'_{(a)} \ge k'_{(i)}$ for $i < a$, d is not one of the a lowest ranked demands of K'. Hence $T(K' + \mathbf{1}_d)_{(i)} = T(K')_{(i)}$ for $i \le a$ and $K + \mathbf{1}_d \succeq K' + \mathbf{1}_d$.

Case: $\boldsymbol{k_{(a)} > k_d}$. Since $k'_d \ge k_d$, and $k_{(i)} = k'_{(i)}$ for $i < a$, the rank r' of k'_d in $T(K')$ is no less than the rank r of k_d in $T(K)$. If $k'_d = k_d$ then $r = r'$, and $T(K + \mathbf{1}_d)_{(i)} = T(K +' \mathbf{1}_d)_{(i)}$ for all $i < a$, except for the possible boundary case $k'_d = k'_{(a-1)} = k'_{(a)}$. In this case $T(K + \mathbf{1}_d)_{(i)} = T(K')_{(i)}$ for all $i < a$, the addition of $\mathbf{1}_d$ to K' yielding an increment of the a-ranked item. The corresponding increment to K is located at its $(a-1)$-ranked item since $k_{(a)} > k'_{(a)} = k'_{(a-1)} = k_{(a-1)}$ and so $T(K + \mathbf{1}_d)_{(i)} = T(K' + \mathbf{1}_d)_{(i)}$ for $i < a - 1$ while $T(K + \mathbf{1}_d)_{(a-1)} > T(K' + \mathbf{1}_d)_{(a-1)}$. Hence $K + \mathbf{1}_d \succeq K' + \mathbf{1}_d$. A similar argument holds if $k'_d > k_d$. □

PROOF OF THEOREM 4. **Case: $\boldsymbol{k_{(a)} < k_{d^*}}$..** Subtracting $\mathbf{1}_{d^*}$ from K leaves its a lowest ranked elements unchanged, and hence $K - \mathbf{1}_{d_*} \succeq K' \succeq K' - \mathbf{1}_d$.

Case: $\boldsymbol{k_{(a)} = k_{d^*}}$. The effect of subtracting $\mathbf{1}_{d^*}$ from K is to decrement the rank a element. Thus if the rank of k'_d in K' is $\le a$, then $T(K' - \mathbf{1}_d)_{(i)} < T(K)_{(i)} = T(K - \mathbf{1}_{d*})_{(i)}$ for one $i \in \{1, \ldots, a-1\}$ and possibly also for $i = a$, with equality for all other $i \in \{1, \ldots, a-1\}$ and the result follows.

If the rank of k'_d in K' is $> a$, we are done except if possibly $k'_{(a)}$ takes the maximum value compatible with $K \succ K'$, namely $k_{d_*} - 1$, for then the first k elements of $T(K - \mathbf{1}_{d_*})$ and $T(K' - \mathbf{1}_d)$ are equal. But then the assumption $\Sigma K \ge \Sigma K'$ implies $\Sigma_{j>a} k'_{(j)} \le 1 + \Sigma_{j>a} k_{d_*}$. We consider the case of equality first; the unequal case follows easily. Since $k'_{(a')} \ge k'_{(a)} = k_{d_*} - 1$ for $a' > a$, either $k'_{(a')} = k_{d_*} - 1 < k_{(a)}$ holds for a' in some set $\{a+1, a+2, \ldots, \widetilde{a}\}$, in which case we are done, or it holds for no such a'. In this latter case, $k'_{(a')} = k_{d_*}$ for all $a' > a$ except for $k'_{(|\mathcal{D}|} = k_{d_*} + 1$. Decrementing the largest of these yields $T(K + \mathbf{1}_{d_*}) = T(K' + \mathbf{1}_d)$ while decrementing any other yields $K + \mathbf{1}_{d_*} \succ K' + \mathbf{1}_d$. □

11. REFERENCES

[1] AT&T's Global Networking Facts. http://www.corp.att.com/globalnetworking.

[2] D. Bertsekas and R. Gallager. *Data Networks*. Prentice-Hall, Englewood Cliffs, NJ, 1992.

[3] R. Bhargava, A. Goel, and A. Meyerson. Using approximate majorization to characterize protocol fairness. In *SIGMETRICS '01*, pages 330–331, New York, NY, 2001.

[4] CAIDA: The Cooperative Association for Internet Data Analysis. http://www.caida.org.

[5] G. R. Cantieni, G. Iannaccone, C. Barakat, C. Diot, and P. Thiran. Reformulating the monitor placement problem: optimal network-wide sampling. In *CoNEXT '06*, pages 5:1–5:12, New York, NY, 2006.

[6] M. T. Chao. A general purpose unequal probability sampling plan. *Biometrika*, 69(3):653–656, 1982.

[7] S. Chaudhuri, G. Das, M. Data, R. Motwani, and V. Narasayya. Overcomng limitations of sampling for aggregation queries. In *ICDE'01*, pages 534–542, 2001.

[8] S. Chen, Y. Fang, and Y. Xia. Lexicographic maxmin fairness for data collection in wireless sensor networks. *IEEE Transactions on Mobile Computing*, 6:762–776, 2007.

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

[10] E. Cohen, N. Duffield, H. Kaplan, C. Lund, and M. Thorup. Stream sampling for variance-optimal estimation of subset sums. In *Proc. 20th ACM-SIAM Symposium on Discrete Algorithms*, 2009.

[11] C. Cranor, T. Johnson, O. Spatscheck, and V. ladislav Shkapenyuk. Gigascope: A stream database for network applications. In *Proc ACM SIGMOD*, June 2003.

[12] A. Dhamdhere, L. Breslau, N. G. Duffield, C. Ee, A. Gerber, C. Lund, and S. Sen. Flowroute: inferring forwarding table updates using passive flow-level measurements. In *IMC 2010*, pages 315–321, 2010.

[13] N. Duffield and C. Lund. Predicting resource usage and estimation accuracy in an IP flow measurement collection infrastructure. In *ACM SIGCOMM Internet Measurement Workshop*, 2003. Miami Beach, Fl, October 27-29, 2003.

[14] N. Duffield, C. Lund, and M. Thorup. Learn more, sample less: control of volume and variance in network measurements. *IEEE Transactions on Information Theory*, 51(5):1756–1775, 2005.

[15] N. Duffield, C. Lund, and M. Thorup. Optimal combination of sampled network measurements. In *Proc. ACM SIGCOMM Internet Measurement Conference (IMC)*, pages 91–104, 2005.

[16] N. Duffield, C. Lund, and M. Thorup. Priority sampling for estimation of arbitrary subset sums. *J. ACM*, 54(6):Article 32, December, 2007. Announced at SIGMETRICS'04.

[17] C. Estan and G. Varghese. New directions in traffic measurement and accounting. In *Proc. ACM SIGCOMM '2002*, Pittsburgh, PA, August 2002.

[18] J. Faffe. Bottleneck flow control. *IEEE Trans. Comm.*, 29(7):954–962, July 1981.

[19] A. Feldmann, J. Rexford, and R. Cáceres. Efficient policies for carrying web traffic over flow-switched networks. *IEEE/ACM Transactions on Networking*, 6(6):673–685, December 1998.

[20] M. Grossglauser and J. Rexford. Passive traffic measurement for IP operations. In K. Park and W. Willinger, editors, *The Internet as a Large-Scale Complex System*. Oxford University Press, 2005.

[21] D. G. Horvitz and D. J. Thompson. A generalization of sampling without replacement from a finite universe. *J. Amer. Stat. Assoc.*, 47(260):663–685, 1952.

[22] M. Hosaagrahara and H. Sethu. Max-min fair scheduling in input-queued switches. *IEEE Transactions on Parallel and Distributed Systems*, 19:462–475, 2008.

[23] F. Kelly. Charging and rate control for elastic traffic. *European Trans. Telecomm.*, 8(1):33–37, 1997.

[24] S. Keshav. *An Engineering Approach to Computer Networking*. Addison-Wesley, Reading, MA, 1997.

[25] J. Kleinberg, E. Tardos, and Y. Rabani. Fairness in routing and load balancing. *Foundations of Computer Science, Annual IEEE Symposium on*, 0:568, 1999.

[26] A. Kumar and J. Xu. Sketch guided sampling - using on-line estimates of flow size for adaptive data collection. In *INFOCOM*, 2006.

[27] N. Kumar, R. Pan, and D. Shah. Fair scheduling in input-queued switches under inadmissible traffic. In *IEEE Globecom*, volume 3, pages 1713–1717, 2004.

[28] S. Leinen. UDP Samplicator. http://www.switch.ch/network/downloads/tf-tant/samplicator.

[29] W. B. Norton. DrPeering.net. http://drpeering.net.

[30] A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: the single-node case. *IEEE/ACM Trans. Netw.*, 1:344–357, June 1993.

[31] M. Pioro, G. Fodor, P. Nilsson, and E. Kubilinskas. On efficient max-min fair routing algorithms. *Computers and Communications, IEEE Symposium on*, 0:365, 2003.

[32] B. Radunović and J.-Y. L. Boudec. A unified framework for max-min and min-max fairness with applications. *IEEE/ACM Trans. Netw.*, 15:1073–1083, October 2007.

[33] A. Ramachandran, S. Seetharaman, N. Feamster, and V. Vazirani. Fast monitoring of traffic subpopulations. In *Proc. Internet Measurement Conference 2008*, pages 257–270, New York, NY, USA, 2008.

[34] V. Sekar, M. K. Reiter, W. Willinger, H. Zhang, R. R. Kompella, and D. G. Andersen. cSamp: A system for network-wide flow monitoring. In *Proc. 5th USENIX NSDI*, San Francisco, CA, Apr. 2008.

[35] K. Suh, Y. Guo, J. Kurose, and D. Towsley. Locating network monitors: Complexity, heuristics and coverage. In *IEEE Infocom'05*, 2005.

[36] M. Szegedy and M. Thorup. On the variance of subset sum estimation. In *Proc. 15th ESA, LNCS 4698*, pages 75–86, 2007.

[37] L. Tassiulas and S. Sarkar. Maxmin fair scheduling in wireless networks. In *in Proceedings of IEEE INFOCOM*, pages 763–772, 2001.

[38] Y. Tillé. *Sampling Algorithms*. Springer, New York, 2006.

[39] J. Vitter. Random sampling with a reservoir. *ACM Trans. Math. Softw.*, 11(1):37–57, 1985.

[40] J. Zhang, X. Niu, and J. Wu. A space-efficient fair packet sampling algorithm. In *Proceedings of the 11th Asia-Pacific Symposium on Network Operations and Management*, APNOMS '08, pages 246–255, Berlin, Heidelberg, 2008.