Predicting Resource Usage and Estimation Accuracy in an IP Flow Measurement Collection Infrastructure

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

Carsten Lund
AT&T Labs—Research
180 Park Avenue
Florham Park, NJ 07932, USA
lund@research.att.com

Categories and Subject Descriptors

C.2.3 [**Computer–Communications Networks**]: Network Operations—*Network monitoring*; C.4 [**Performance of Systems**]; G.3 [**Probability and Statistics**]

General Terms

Measurement, Performance, Theory

Keywords

Sampling, Estimation, Variance, Bandwidth

ABSTRACT

This paper describes a measurement infrastructure used to collect detailed IP traffic measurements from an IP backbone. Usage, i.e, bytes transmitted, is determined from raw NetFlow records generated by the backbone routers. The amount of raw data is immense. Two types of data sampling in order to manage data volumes: (i) (packet) sampled NetFlow in the routers; (ii) size-dependent sampling of NetFlow records. Furthermore, dropping of NetFlow records in transmission can be regarded as an uncontrolled form of sampling.

We show how to manage the trade-off between estimation accuracy and data volume. Firstly, we describe the sampling error that arises from all three types of sampling when estimating usage per traffic class: how it can be predicted from models and raw data, and how it can be estimated directly from the sampled data itself. Secondly, we show how to determined the usage of resources—bandwidth, computational cycle, storage—within the components of the infrastructure. These two sets of methods allow dimensioning of the measurement infrastructure in order to meet accuracy goals for usage estimation.

1. INTRODUCTION AND MOTIVATION

The collection of network usage data is essential for the engineering and management of communications networks. Until recently, the usage data provided by network elements (e.g. routers) has been coarse-grained, typically comprising aggregate byte and packet counts in each direction at a given interface, aggregated over time windows of a few minutes. However, these data are no longer sufficient to engineer and manage networks that are moving beyond the undifferentiated service model of the best-effort Internet. Network operators need more finely differentiated information on the use of their network. Examples of such information include (i) the relative volumes of traffic that use different protocols or applications; (ii) traffic matrices, i.e., the volumes of traffic originating from and/or destined to ranges of Internet Protocol (IP) addresses or Autonomous Systems (AS's); (iii) the aggregate statistics of packet and byte volumes and durations of user sessions. Such information can be used to support network management, in particular: traffic engineering, network planning, peering policy, customer acquisition, usage-based pricing, and network security; some applications are presented in details in [2, 11, 12]. An important application of traffic matrix estimation is to efficiently redirect traffic from overloaded links.

A prototype Traffic Analysis Platform (TAP) has been developed in order to achieve these goals. It allows for the collection and processing of flow measurements from a wide area backbone network. The main challenge for the TAP is the immense amount of backbone traffic, and hence the proportionately immense amount of flow measurements to be collected. The TAP meets this challenge with a distributed architecture and extensive use of sampling and aggregation at multiple measurement locations. Sampled NetFlow (see Section 2.5.2) is configured on the routers, reporting aggregate measured from sampled packet streams. The resulting flow records are subject to aggregation and sampling on their passage through the measurement infrastructure. A form of non-uniform sampling introduced in [6] (here called smart sampling; see Section 2.5.4) of the completed NetFlow records is implemented at collection points. Inherent in sampling is the consequence that network usage is estimated rather than known exactly. However the sampling methods, in particular smart sampling, are optimized in order to yield estimates of minimal variance subject to a given constraint on resource usage in the TAP.

This paper provides a detailed description of the TAP, the sampling methods that operate in it, and established the relationship between resource usage in the measurement infrastructure and statistical accuracy of usage estimates from the sampled data. This enables the dimensioning of the measurement infrastructure in order to meet accuracy goals. In doing this, we build on and extend prior work on smart sampling [6] and on the statistical properties of packet sampled flows [7]. In particular:

- We derive bounds for the sampling variance of usage estimates due to the cumulative effect of packet Sampled Net-

IMC'03, October 27–29, 2003, Miami Beach, Florida, USA.

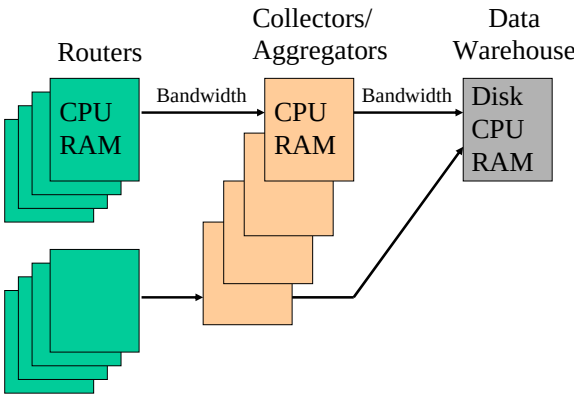

Figure 1: TAP Architecture and Resources

Flow, smart sampling, and transmission loss, in terms of the sampling parameters and simple traffic characteristics. Furthermore, we show how this variance of a given usage estimate can itself be estimated on the fly from the sampled data.

- We derive bounds and estimates for the usage of different resource in the TAP architecture in terms of the sampling parameters and traffic flow characteristics. These bounds can be used to predict resource usage from models or traces, or to predict the effect on resource usage of a change in parameter settings.

The purpose of this paper is to describe the set of analytical tools that enable planning of resources in a TAP measurement infrastructure, that enable the correct setting of sampling parameters and dimensioning of resource, compatible with accuracy goals for the estimation of network usage. The paper is organized as follows. Section 2 describes the TAP architecture, and the sampling operations that take place within it. Section 3 describes the model for sampling process and relevant traffic properties.

With this setup, the main work of the paper is to show how the selection of sampling parameters determines both the variance of usage estimates and volumes of samples selected. Section 4 contains the main results on bounding sampling errors. Section 5 shows how to estimate the rate of production of sampled flow records from a router. Section 6 establishes bounds and estimates for the rate of production of smart sampled records, and the rate at which aggregates of these are formed. Section 7 reports some examples of using these methods with NetFlow data. We conclude in Section 8. Mathematical proofs are deferred to an Appendix.

2. THE TAP ARCHITECTURE

2.1 Components

The components of the TAP architecture are shown in Figure 1:

- *Routers:* these collect raw NetFlow records on the traffic that they route. These are exported in UDP packets to a local collection server.

- *Collector/Aggregators:* these have three tasks. First, they receive the raw NetFlow records from the routers and write them to local disk. Second, they aggregate raw records and create various higher level aggregates. Third, the aggregates are shipped to a central Data Warehouse. Collection servers are placed at major geographical locations in the backbone network. This makes the architecture scalable and reduced bandwidth consumption by the transmission of raw flows in the backbone.

- *Data Warehouse:* this stores the aggregate data, which can generate reports and field ad-hoc queries from an end-user community including engineering, product management and research organizations.

2.2 Resources

The resources of these components, and their usage, is as follows. The load on transmission and CPU resources at a given component is proportional to the rate which flow records arrive at it. During aggregation, memory usage is proportional to the number of distinct flow keys that present during the aggregation period. At the end of each such period, the aggregated flows are written to disk; thus the rate of consumption of disk storage space is proportional to the number of distinct aggregate flow keys, divided by the duration of the aggregation period. A substantial part of the the work of this paper is to predict the usages of these resources through a combination of measurement and analysis.

2.3 Aggregation and Queries

The software running on the collection servers was required to be efficient and flexible in the following sense. It needs to be efficient, since it needs to process an immense amount of data. It needs to be flexible since, due to changes in end user requirements and the need to field ad-hoc queries, it is not possible to determine in advance a set of aggregates that would support all potential queries. Indeed, the requirement for flexibility is a reason that aggregate records should not be formed at the router.

The flexibility of TAP is achieved by implementation of a domain specific language `tapquery` that allows the users to write high-level queries that the system then will run on the collection servers. The language contains constructs that allow:

1. Definition of flow keys and outputs for aggregation. For example, using source and destination IP address as the key and flow bytes as output yields the host-to-host traffic matrix.

2. Joining the raw flow data with external data sources. For example, joining with a table of IP address block used by user groups yields traffic usage by user group.

3. Filtering. Some applications focus on a subset of traffic, e.g, filtering by TCP/UDP port number can be used to restrict scope to traffic using specific protocols or applications. Filtering by interface also restricts focus to traffic associated with a given user, or a given peer,

4. Smart sampling. As further explained in Section 2.5.4, smart sampling selects a subset of NetFlow records as input for queries or aggregation, and performs appropriate renormalization of measured usage in order that usage estimates are unbiased.

5. Correction for data loss in the raw measurement stream. NetFlow records may be lost in transmission from router to collection server, or for other reasons. For example, a surge in NetFlow records may occur during a denial of service attack due to the presence of a large number of short flows with spoofed source IP address. Correction enables the tracking of actual network usage even if NetFlow records are lost.

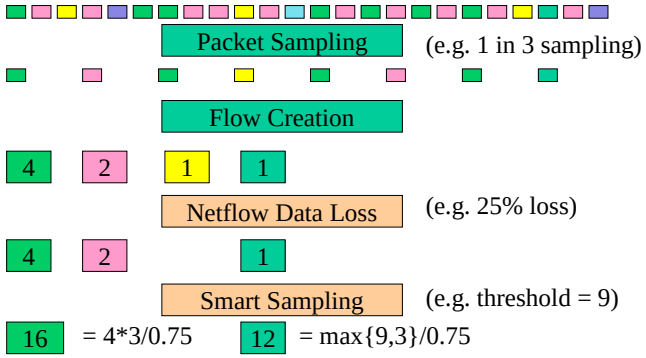

Figure 2: TAP Sampling Operations. See Section 2.5.6 for description of the examples.

Finally the aggregation systems allows computing multiple queries at the same time, dynamically add, delete or change queries on the fly, and dynamically use updated external data sources.

2.4 NetFlow Records

The fundamental measurements collected by routers in the TAP architecture are flow records. We review how these are formed. An IP flow is a set of packets, observed in the network within some time period, and that share some common property known as its key. The fundamental example is that of so-called "raw" flows: a set of packets observed at a given network element, whose key is the set of values of IP and other protocol header fields. A router keeps records on active flows passing through it. When a packet arrives at the router, the router performs a lookup on its cache of currently active flows, to determine if a flow is active for the packet's key. If not, is instantiates a new record for the packet's key. The record include counters for packets and bytes that are updated according to each packet that matches the key.

When the flow is terminated, its record is flushed for export, and the associated memory released for use by new flows. A router will terminate the flow if any one of a number of criteria are met, including (i) timeout: the interpacket time within the flow will not exceed some threshold; (ii) protocol: e.g., observation a FIN packet of the Transmission Control Protocol (TCP) [18] that terminates a TCP connection; (iii) memory management: the flow is terminated in order to release memory for new flows; (iv) aging: to prevent data staleness, flows are terminated after a given elapsed time since the arrival of the first packet of the flow. Flow definition schemes have been developed in research environments, see e.g. [1, 5], and are the subject of standardization efforts [16]. Flow records typically include the properties that make up flows defining key, its start and end times, and the number of packets and bytes in the flow.

In the TAP architecture, routers generate flow records from a sampled subset of the packet stream using Sampled NetFlow [4]. In the following section we motivate and describe this and other sampling methods used in the TAP architecture.

2.5 Sampling in the TAP Architecture

Sampling is employed in the TAP architecture in order to control resource usage. The set of sampling operations available is illustrated in Figure 2. (The numbers relate to an example that is worked in Section 2.5.6). The top line of the figure illustrates a sequence of packets incident at the router. During the operation of sampled NetFlow, a proportion of these packets are selected and flow records constructed from them. Dropping of NetFlow records in transit from the router to the aggregator can be viewed as another type of sampling. In the aggregator, smart sampling is applied to the flow records themselves. We describe each of these sampling operations in the following paragraphs.

2.5.1 Renormalization of the Samples

Sampling progressively thins the information that flows through TAP by discarding packets and flows. In order to obtain an unbiased estimate of usage in the original traffic stream, it is necessary to compensate for the effects of this discard by renormalizing the usage that survives sampling. Specifically, the usage represented in each surviving packet or flow must be divided by the probability of its selection. We shall see in Section 4 that this yields an unbiased estimate for the usage. So-called unequal probability sampling is frequently used in sample design in order to sample preferentially from amongst larger components of a population; see [15].

For each of TAP's sampling operations described in the following paragraph, we also describe the renormalization that is applied to usage data that survives sampling.

2.5.2 Sampled NetFlow

In order to perform flow cache lookups at line rate, a high end router would need to be equipped with large amounts of fast—and hence expensive—memory, in order to maintain records on the expected numbers of concurrently active flows. In order to limit the frequency of flow cache lookups, sampled NetFlow forms flow records from a substream of packets. In current implementations this is performed by periodically selecting every N^{th} packet. Other potential implementations include pseudorandom independent selection of packets with probability $1/N$, and randomized selection driven by the entropy of the packet contents itself; see [9]. Whatever the implementation, in order to form unbiased usage estimates, the usage attributed to each selected packet of size b is multiplied by N (the reciprocal of the selection probability), yielding Nb.

A different approach to packet sampling has been taken in [10], where prospective new flow cache entries are sampled, with only those selected being instantiated. This favors the recording of longer flows, while suppressing recording of short flows that contribute little to usage. Other work has considered adapting packet sampling rates in order to maintain estimation accuracy; see [3].

2.5.3 Dropping Flow Records

Routers export NetFlow records to the collector using UDP. Since UDP possesses no ability to detect or correct for losses in transmission, flow records may be lost during periods of congestion. The flow records contain a sequence number that enables the loss to be detected by the collector, and to determine the rate at which loss occurs. If the average rate of successful transmission is q, the number of bytes reported is normalized through division by q.

2.5.4 Smart Sampling

In the collector, flow records are selected by a form of non-uniform sampling called smart sampling. Smart sampling is controlled through a parameter known as the *sampling threshold*, which we shall denote by z. In smart sampling, a flow record representing a flow of x bytes is sampled with probability $p_z(x) = \min\{1, x/z\}$. Flows of size greater than the threshold z are always selected, while smaller flows are selected with probability proportional to their size.

The motivation for smart sampling comes from the empirical fact that flow sizes have a heavy tailed distribution; [13]. In this case,

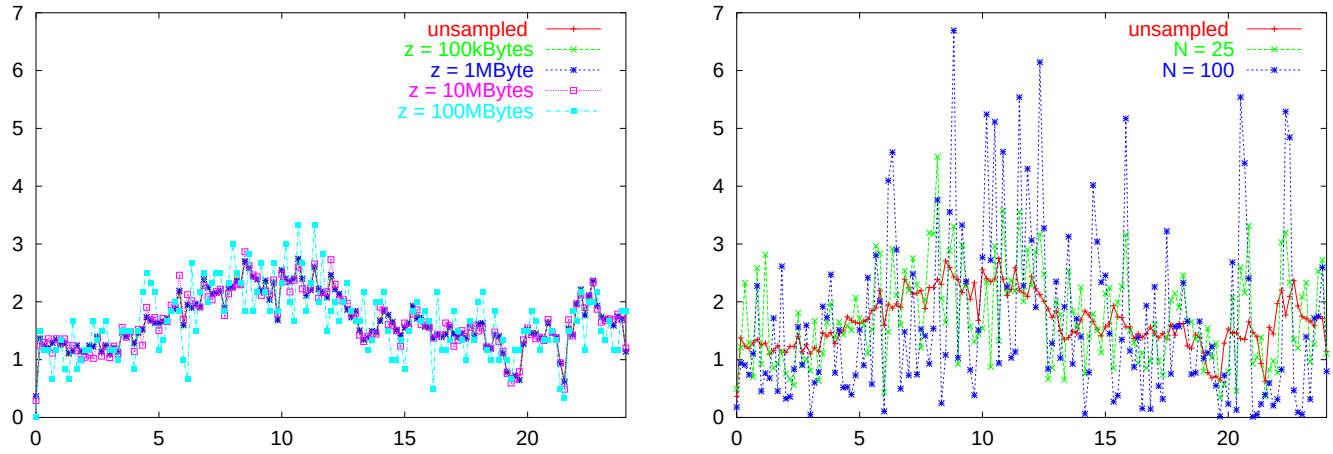

Figure 3: ESTIMATED 10 MINUTE BYTE RATES FOR NNTP TRAFFIC OVER 24 HOUR PERIOD: smart sampling (left) and uniform sampling (right)

sampling with a uniform distribution over flow sizes is problematic for usage estimation, since the estimates are very sensitive to omission of a single large flow. Smart sampling avoids this problem by always selecting large flows. This approach was proposed in [6, 7]. The probability p_z represents an optimal trade-off between the variance v of usage estimation, and expected number n of flows sampled in the following sense. Although in principle any non-zero size-dependent probability $p(x)$ may be used for flow sampling, p_z is distinguished by the property that it minimizes $v + z^2 n$, regardless of the actual flow lengths. It is worth noting that, even in the composite sampling scheme considered in this paper, once the packet sampling period N is specified, use of flow sampling probability p_z is still optimal in the above sense.

More generally, one can consider the trade-off between packet sampling and flow sampling. From Section 2.5.2 it is clear that some amount of packet sampling is required at the router. In experiments it is found that when further sampling is required, smart sampling yields smaller bandwidth for a given variance of usage estimates [7]. This is because it can capitalize on the compression inherent in the formation of flow records, as compared with reports on the constituent packets.

In order to form an unbiased estimate of the original usage, a flow that reports x bytes, is normalized through division by the selection probability, i.e., its contribution to the usage estimate is $x/p_z(x) = \max\{x, z\}$ bytes. Thus flows whose size exceeds the threshold z are (always) reported unaltered. Flows whose size is less than z, have their size reported at z, if they survive sampling.

2.5.5 *Comparing Smart and Uniform Sampling*

Smart sampling allows vastly improved estimation accuracy, as compared with uniform sampling, for a given volume of collected flow records. As an illustrative application, we compare the efficacy of the two sampling methods in estimating usage in classes of traffic. Raw NetFlow records were collected from a router over a 24 hour period. The task is to estimate byte usage by network news (nntp) traffic, as identified by application port number. This traffic comprised roughly 0.55% of all observed traffic. Smart sampling and uniform sampling were applied to each of the raw NetFlow records for a range of sampling parameters. Figure 3 shows the estimated byte usage for smart sampling (left) and uniform sampling (right), expressed as an average byte rate over 10 minute intervals. Table 1 shows the effective sampling periods (i.e. the reciprocal of the average rate at which flows are sampled), and the relative errors between estimated and actual rates, maximized over the set of 10 minute intervals.

In Figure 3(left) the points for smart sampling with threshold z up to 1MByte are virtually indistinguishable from the true values (i.e. those with no sampling). With this threshold, about 1 in 127 flows are sampled, and the relative error is about 0.02. With uniform sampling at a comparable flow sampling rate, 1 in $N = 100$, the relative error is 3.1, i.e. over 150 times larger. This is reflected in Figure 3(right), in which the estimated byte rates can differ greatly from the true rates. By contrast, when smart sampling threshold z = 10MBytes, i.e., sampling a little less than 1 in 1,000 flows on average, the systematic variation in byte rate can still be clearly discerned. An example showing the accuracy of smart sampling in estimating per address usage is described in [6].

2.5.6 *Composition of Sampling and Renormalization*

Although sampling in TAP occurs in a specific order (packet sampling, followed by flow dropping, followed by smart sampling) the corresponding renormalizations may be applied in any order without biasing usage estimates. In the TAP architecture, renormalization for packet sampling is applied first, followed by smart sampling, finishing with flow dropping. The flow dropping normalization is applied last because the average drop rate is determined only after flow records have been aggregated.

Method	Parameter	Flows Sampled	Period	Rel. Error
No sampling		328,198,000	1	0
Smart	z = 100kBytes	13,468,500	24	0.0028
Smart	z = 1MByte	2,574,900	127	0.020
Smart	z = 10MBytes	326,222	1006	0.23
Smart	z = 100MBytes	33,117	9910	0.81
Uniform	$N = 25$	13,127,900	25	1.2
Uniform	$N = 100$	3,281,980	100	3.1

Table 1: COMPARISON OF SAMPLING METHODS: method; parameters; flow records sampled; proportion of flow records sampled; maximum relative error for 10 minute average rates.

We illustrate with the examples in Figure 2. The packet sampling period is $N = 3$, the packet transmission probability is $q = 0.75$, and the smart sampling threshold is $z = 9$. For simplicity we set all packets to have the nominal size of 1 byte.

The first sampled flow record has $b = 4$ bytes, and survives transmission. It enters smart sampling reporting $Nb = 12$ bytes. This exceeds the smart sampling threshold of $z = 9$, and hence this flow is smart sampled with probability 1, reporting 12 bytes. Finally, the reported bytes are normalized through division by q, yielding 16 bytes.

The second sampled flow record has $b = 2$ bytes, and survives transmission. It enters smart sampling reporting $Nb = 6$ bytes. Since this is less than z, it is smart sampled with probability $Nb/z = 2/3$. In happens to be discarded.

The third sampled flow record has $b = 1$ byte, and happens to be lost in transmission.

The fourth sampled flow has size $b = 1$ byte, and is transmitted successfully. It enters smart sampling reporting $Nb = 3$ bytes. Since this is less than z, it is smart sampled with probability $Nb/z = 1/3$. It happens to be selected, and exits smart sampling reporting $\max\{z, Nb\} = 9$ bytes. Finally, the reported bytes are normalized through division by q, yielding 12 bytes.

The total bytes reported is 28; the total number of bytes of the original packet stream (top row) was 24.

3. MODELING SAMPLING PROCESSES

The aim of this section is to model the effect of sampling upon usage measurements. In doing so we regard the set of packets and flows sizes of the traffic as *deterministic* quantities, representing the actual usage that is to be estimated. The only randomness that enters is due to the sampling itself. In applications we wish to estimate the usage for each traffic class of interest. The statistical properties of the underlying traffic, which may be very complex, do not enter the estimates.

3.1 Model for Sampled NetFlow

Within the functional requirement of sampling packets at a given rate, different implementations are possible. In Sampled NetFlow as provided by Cisco [4] packets are selected periodically, i.e., every N^{th} packet is selected for some period N. Another possibility is to sample packets independently with probability $1/N$–e.g. as performed by sFlow [14], then compile flow records from the sampled packets. To what extent do the implementation details have ramifications for modeling the sampling process?

Periodic sampling introduces correlations into the sampling process: when a packet is selected, none of the following $N-1$ packets are selected. Although this does not bias against selection of any one packet, it can bias against selection of multiple packets from short flows. We do not believe this effect would be important for sampling from high speed links that carry many flows concurrently. In this case, successive packets of a given flow would be interspersed by many packets from other flows, effectively randomizing packet selection from the given flow. While such randomization may not be effective at lower speed routers carrying fewer flows (e.g. edge routers), packet sampling is not expected to be crucial for flow formation in this case. For these reasons, we will model the sampling of packets from a given flow as being independent.

3.2 Model for Dropping of Flow Reports

We assume that flow records are transmitted independently with some probability q. Thus, we can view record loss as equivalent to independent sampling with probability q. Note that, in principle, more complex patterns of dependent loss amongst flow records could also be detected from the received sequence numbers, and an appropriate model constructed. We do not pursue this here.

3.3 Model for Smart Sampling

We assume that, conditioned on the set of packet and flows presented in the traffic, the selections of flows during smart sampling are mutually independent. This property is sometimes called conditional independence. Conditionally independent selection occurs when an independent random variate ω in $[0, 1]$ is generated for each flow, a flow of size x being selected if $\omega \leq p_z(x)$. This manner of sampling can be implemented using a pseduorandom generator (such as [17]) for the ω. Note that the potentially complex statistical properties of the the traffic process— such as heavy tailedness of flow lengths and correlations amongst flows and packets—do not play a role because we condition on the single "sample-path" of traffic that is actually present. (On the other hand, if one averaged over a distribution of sample-paths that exhibited dependence between flow length, flow selection would be dependent).

An alternative approach to random selection is to use the flow sizes themselves as a source of randomness; see [8]. This is computationally simpler, but does incur some dependence between selection of different flows. However, similar arguments to those we made above for periodic packet sampling lead us to expect that dependence will be weak when considering flows of a given key. This was found to be the case in experiments.

3.4 Sparse Flows and Splitting

Packet sampling can increase the number of measured flows. Given a sampling period N and a flow interpacket timeout T, we say that a given original flow of packets is *sparse* if the typical time between sampled packets exceeds T. In this case, a single original flow may give rise to multiple flow records, each sampled packet giving rise to one measured flow record. To see more precisely when this can happen, consider an original flow comprising n packets distributed over an interval of duration t. The typical time between sampled packets is tN/n, thus sparseness requires than $tN/(nT) > 1$. It also requires that there is typically more than one sampled packet, i.e., $n/N > 1$. Combining, we can say that the threshold for sparseness is crossed when

$$\frac{t}{T} > \frac{n}{N} > 1. \tag{1}$$

From these conditions, we see that sparseness is most likely to arise for flows containing many packets occurring with relatively low frequency. In experiments, it is found that streaming and multimedia applications generate sparse flows at what may be reasonable settings for sampled flow measurement: sampling period $N = 100$ and flow interpacket timeout $T = 30s$. See [7] for further details.

In this paper our interest in sparseness lies in understanding its impact, if any, on the variance of usage estimates, and the volume of flow records. Although splitting may increase or decrease the estimation variance, a simple bound we obtain is unaffected, regardless of splitting. In order to calculate the effect on the volume of measured flows, we shall need to adopt a particular model of the distribution of packets in the flow.

4. SAMPLING ERROR IN USAGE ESTIMATES

Reduction by sampling of the volume of sampled data comes at the cost of inherent uncertainty over the estimates of network usage derived therefrom. Smart sampling has been tailored to optimize the trade off between sample volume and estimator accuracy, and to mitigate against the high variability of estimation that would occur if flow records were uniformly sampled.

4.1 Bounds on the Sampling Error

We aim to estimate the total usage $X = \sum_{i=1}^{n} x_i$ from n flows of sizes $x_1, \ldots, x_n$, for example, flows in a given traffic class of interest. Each flow i in comprises m_i packets of sizes $b_{i1}, \ldots, b_{im_i}$; hence $x_i = \sum_{j=i}^{m_i} b_{ij}$. We construct an estimator $\widehat{X}$ of the usage X according to the sampling operations and normalizations described in Section 2.5, as modeled in Section 3. We will use random indicators (variables taking the value 0 or 1) to write $\widehat{X}$ as a random sum over all packets and flows. The variance of $\widehat{X}$ derives entirely from the statistical properties of these indicators. The quantities n, m_i and b_{ij} are considered fixed in any given estimation problem.

Estimation takes the following form in each stage of sampling. An object (a packet or a flow) of some size $\widehat{x} > 0$ is selected with some probability $p(\widehat{x}) > 0$ that may depend on $\widehat{x}$. The size $\widehat{x}$ may itself be a random quantity arising from an earlier stage of sampling. Let w be an indicator random variable, conditionally independent of $\widehat{x}$, that takes the value 1 with probability $p(\widehat{x})$. Then we form an estimate $\widehat{y} = \widehat{x} \cdot w/p(\widehat{x})$, i.e., by multiplying with the random quantity $w/p(\widehat{x})$. If the object is not selected, $w = 0$, i.e., usage which is not sampled makes no contribution to usage estimates. On the other hand, if the object is selected, $w = 1$, the the contribution $\widehat{x}$ to usage is scaled up by a factor $1/p(\widehat{x})$ relative to the actual usage. But for a given value of $\widehat{x}$, w/p has expectation 1, and hence $\widehat{y}$ is an unbiased estimator of $\widehat{x}$ in the sense that the its conditional expectation obeys $\mathsf{E}[\widehat{y} \mid \widehat{x}] = \widehat{x}$.

We also need to treat the special case $\widehat{x} = 0$ which arises when the object was not selected in some previous stage of sampling. In this case we want to have $\widehat{y} = 0$ too. However, it is useful to have the definition $\widehat{y} = \widehat{x} \cdot w/p(\widehat{x})$ work transparently in the calculations. A potential problem arises if $p(0) = 0$; this happens for smart sampling but not for packet sampling. A general approach is to assume that $x/p(x)$ is continuous from the right at $x = 0$. (This is true for smart sampling). Then when $p(0) = 0$ we can define $x/p(x)$ at $x = 0$ by continuity. As a result $\widehat{y} = 0$ as required, since either $p(0) > 0$ (in which case $x/p(x) = 0$ at $x = 0$), or $p(0) = 0$ (in which case w is 0 with probability 1 when $x = 0$).

In preparation for analyzing the statistical properties of the usage estimates, we use the scheme just described to write the usage estimates in terms of the underlying packet and flow sizes, and the indicator random variables for sampling.

- *Packet Sampling:* The total estimated bytes from packets from flow i are $N \sum_{j=1}^{m_i} u_{ij} b_{ij}$ where the u_{ij} are mutually independent indicator random variables taking the value 1 if packet j of flow i is sampled. Thus $u_{ij} = 1$ with the probability that a packet is sampled, namely $1/N$.

- *Flow Record Loss:* The application to loss of flow records is achieved through further multiplying by v_i/q, where the v_i are mutually independent indicator random variables taking the 1 if flow record i would survive transmission, i.e., with probability q. The resulting unbiased estimate of x_i is

$$\widehat{x}_i = Nq^{-1} \sum_{j=i}^{m_i} v_i u_{ij} b_{ij}. \tag{2}$$

- *Smart Sampling:* the application to smart sampling is complicated slightly by the fact, mentioned in Section 2.5.6, that usage renormalization to compensate for the loss of flow records is performed only after smart sampling. The size reported in the normalized sampled flow record presented for smart sampling is $q\widehat{x}_i$. Given a smart sampling threshold z, then assuming no splitting of flows, the record survives smart sampling with probability $p_z(q\widehat{x}_i) = p_{z/q}(\widehat{x}_i)$. If so, its reported size, after normalization with q, is $q^{-1} \max\{z, q\widehat{x}_i\} = \max\{zq^{-1}, \widehat{x}_i\}$; see Section 2.5.4. Thus, estimated usage arising from smart sampling of the measured flow (if any) produced by sampling the packets of original flow i is

$$\widehat{y}_i = w_i \max\{zq^{-1}, \widehat{x}_i\}, \tag{3}$$

where the w_i are mutually independent indicator random variables taking the value 1 if flow record i would be selected during smart sampling, i.e., with probability $p_{z/q}(\widehat{x}_i)$.

The final estimate $\widehat{X}$ of total usage resulting from all packets is obtained by summing over all original flows i:

$$\widehat{X} = \sum_i \widehat{y}_i = q^{-1} \sum_i w_i \max\{z, N \sum_{j=1}^{m_i} v_i u_{ij} b_{ij}\} \tag{4}$$

In order to determine the variance of $\widehat{X}$ we apply a conditioning equality for variances for each stage of sampling. Let A and B be random variables and define the conditional variance of A given B by

$$\mathsf{Var}(A|B) = \mathsf{E}[(A - \mathsf{E}[A \mid B])^2 | B]; \tag{5}$$

see, e.g., Problem 8 of Chapter 1 in [20]. Then the conditional and unconditional variance of A are related by

$$\mathsf{Var}(A) = \mathsf{E}[\mathsf{Var}(A \mid B)] + \mathsf{Var}(\mathsf{E}[A \mid B]) \tag{6}$$

Consider again the unbiased estimate $\widehat{y} = \widehat{x} \cdot w/p$ of $\widehat{x}$. Then one can separate out the component of the variance of $\widehat{y}$ that is due to the estimation step from the inherent variance of $\widehat{x}$ as follows:

LEMMA 1. $\mathsf{Var}(\widehat{y}) = \mathsf{E}[\widehat{x}^2 \dfrac{1 - p(\widehat{x})}{p(\widehat{x})}] + \mathsf{Var}(\widehat{x})$.

Applying this Lemma to each sampling stage in the formation of $\widehat{X}$, we obtain the following results for the variance of $\widehat{X}$, which are proved, along with Lemma 1, in the Appendix. Let $x_{\max}$ denote the maximum flow size and $b_{\max}$ the maximum packet size.

THEOREM 1. *Assume independent packet sampling with period N and smart flow sampling with threshold z.*

(i) *$\widehat{X}$ is an unbiased estimator of X.*

(ii) $$\mathsf{Var}\,\widehat{X} = \sum_{i=1}^{n} \mathsf{E}[\widehat{x}_i \max\{zq^{-1} - \widehat{x}_i, 0\}] + \frac{1-q}{q} \sum_{i=1}^{n} (\sum_{j=1}^{m_i} b_{ij})^2 + \frac{N-1}{q} \sum_{i=1}^{n} \sum_{j=1}^{m_i} b_{ij}^2.$$

(iii) $\mathsf{Var}\,\widehat{X} \le q^{-1} X\,(z + (1-q)x_{\max} + (N-1)b_{\max})$.

(iv) *When flows can be split, the same bound (iii) holds for the variance $\mathsf{Var}\,\widehat{X}'$ of corresponding usage estimate $\widehat{X}'$.*

4.2 Interpretation of Theorem 1

The expressions in Theorem 1(ii) and (iii) are sums over contributions due to the different types of sampling: the first from smart sampling, the second from loss of flow records, and the third from packet sampling. We now interpret the meanings of the terms.

Traffic Usage X (GB)	Average Flow Size x (MB)	Maximum Packet Size, Bytes	Sampling Threshold z (MB)	Packet Sampling Period N	Report Loss Rate 1-q	Smart Sampling Standard Error	Packet Sampling Standard Error	Flow Loss Standard Error	Total Standard Error
1	1	1500	1	500	0%	3.16%	2.74%	0.00%	4.18%
10	1	1500	1	500	0%	1.00%	0.87%	0.00%	1.32%
0.1	1	1500	1	500	0%	10.00%	8.65%	0.00%	13.22%
1	1	1500	10	500	0%	10.00%	2.74%	0.00%	10.37%
10	1	1500	1	500	0%	1.00%	0.87%	0.00%	1.32%
1	1	1500	1	5000	0%	3.16%	8.66%	0.00%	9.22%
1	1	1500	1	50	0%	3.16%	0.86%	0.00%	3.28%
1	1	1500	1	500	10%	3.16%	2.88%	1.05%	4.41%
1	1	1500	1	500	50%	3.16%	3.87%	3.16%	5.91%
1	1	1500	1	500	90%	3.16%	8.65%	9.49%	13.22%

Table 2: SAMPLING STANDARD ERRORS: broken down for packet sampling, flow loss, and smart sampling.

4.2.1 Computational Issues

Observe that when $q = 1$ (no transmission loss) the bound (iii) needs only two broad characteristics of the traffic: the total volume X to be estimated, and the maximum packet size $b_{\max}$. The latter can in turn be bounded above by the Maximum Transmission Unit of the network under measurement. By comparison, the exact expression (ii) requires knowing the characteristics of each flow; it requires far more detailed measurements (including packet sizes, which may not be available), and it computationally more intensive. We shall see in Section 7 that the bound in fact gives a very close approximation to the actual variance in cases examined.

4.2.2 The Effect of Sparse Flows

Although the upper bound Theorem 1(iii) is unaffected by splitting of sparse flows, the expression (ii) is generally impacted. Variance due to flow loss decreases, since flows are split across multiple reports. Variance due to packet sampling is unchanged. The smart sampling variance may increase or decrease.

4.2.3 Large Flows and Estimator Variance

Note that when X is dominated by the contribution of one very large flow, the last term in the variance bound may be close to $(1-q)/q$. The the standard error may be quite large if the dropping rate is also large. This is not surprising: smart sampling was applied in the collector precisely so as to mitigate against the effects on accuracy of uniform sampling of flow records whose reported sizes have a heavy-tailed distribution. For this reason, we recommend that the bandwidth for transmission of raw flow records and computational resources on the collector be sufficiently large to accommodate the records without loss under normal operation. Later in this paper, we provide estimates for the required bandwidth.

4.2.4 Comparing Variance: Packet, Smart Sampling

Assuming that our recommendation to dimension collection infrastructure for no report loss is followed, the bound of Theorem 1(iii) takes a simple form: $\mathsf{Var}\, X \leq X(z + (N-1)b_{\max})$. From this bound, we expect the ratio of variance due to smart sampling to that due to packet sampling is about $z/(Nb_{\max})$. Thus for typical values $b_{\max} = 1,500$ Bytes, $N = 100$, the smart sampling variance exceeds the packet sampling variance only when $z > 150$ kBytes.

4.2.5 Resampling of Aggregates

In the TAP infrastructure, smart sampled flow records may be aggregated over time, and the resulting aggregates subject to further smart sampling with a threshold z_2. What is the effect on estimator variance? Without aggregation, it was shown in [7] that composition of n smart sampling stages with thresholds $z_1 \ldots, z_n$ is equivalent to a single smart sampling with threshold $\max_{i=1}^{n} z_i$. With aggregation, we have found no such simple relation. However, an application of Lemma 1, together with the bounding methods used to establish Theorem 1(iii), show that the additional variance in $\widehat{X}'$ introduced is bounded above by $q^{-1}Xz_2$.

4.2.6 Application and Examples

The bound of Theorem 1(iii) is independent of any distribution details of the flow sizes themselves. This enables us to construct simple bounds on estimator variance in term of *average* properties of flows. We have tabulated the bound for the standard error $\sqrt{\mathsf{Var}\,\widehat{X}}/X$ in Table 2 for the case of no report loss: $q = 1$. For comparison, we include a version of the same bound for $q < 1$, but based on the assumption that all flows have the same length. As remarked in Section 4.2.4, the actual variance due to flow report loss may be much larger due to the random selection of long flows.

4.3 Composed Sampling and Estimating Variance from Measurements

Theorem 1 bounds estimator variance in terms of the packet and flow sizes of the unsampled data. But in practice we will want to determine estimator variance directly knowing only the sampled data values. For example, only usage of packets sampled in the router will reach the collector. However, in practice we wish to estimate the variance of our usage estimates directly from the data that survives sampling. The TAP architecture encompasses multiple levels of sampling (packet sampling, report loss, smart sampling) and aggregation (packets into flows, aggregation of smart sampled flows) and it is desirable to estimate the total variance contributed by each stage of sampling and aggregation.

4.3.1 Estimating Sampling Variance

As before, consider sampling an objects of size x with probability $p(x)$, and let w be the indicator random variable for sampling the object, i.e., w is conditionally independent of x and takes the value 1 with probability $p(x)$. We have seen that $\widehat{x} = x \cdot w/p(x)$ is an unbiased estimator of x. It is straightforward to show (see [6]) that $\widehat{x}$ has variance $v(x) = x^2(1-p(x))/p(x)$. Similarly to before, we can form an unbiased estimator of $v(x)$ by renormalizing the quantity $v(x)$ for those objects that are selected during sampling. Thus (see [6])

LEMMA 2. *$\widehat{v}(x) = v(x) \cdot w/p(x)$ is an unbiased estimator of $v(x)$: $\mathsf{E}[\widehat{v}(x)] = v(x)$.*

Note that the variance estimator is *not* simply a sum of $v(x)$ over sampled objects. This would underestimate the variance. It is necessary to divide by the sampling probability $p(x)$.

The two applications of Lemma 2 in this paper are:

- *Uniform Sampling:* Here $p(x) = 1/N$. We write the variance of $\widehat{x}$ as $v_u(N, x) = x^2(N-1)$. The corresponding variance estimator is $\widehat{v}_u(N, x) = x^2N(N-1)$.
- *Smart Sampling:* $p(x) = p_z(x)$. We write the variance of $\widehat{x}$ as $v_s(z, x) = x \max\{z - x, 0\}$. The corresponding variance

estimator is $\widehat{v}_s(z,x) = z\max\{z-x,0\}$. As one might expect, there is no sampling variance associated with objects of size associated with objects larger than the sampling threshold: $v_s(z,x) = 0$ for $x \geq z$.

4.3.2 Estimated Variance in Composed Sampling

We now show how to estimate the incremental variance incurred by each stage of sampling from the objects that are selected at that stage. As before, consider the unbiased estimator $\widehat{y} = \widehat{x} \cdot w/p(\widehat{x})$ arising from sampling an object as size $\widehat{x}$, which is itself a randomly sampled quantity. The incremental contribution to sampling variance due to composed sampling is described by the following result, which can be regarded as combination of Lemmas 1 and 2.

LEMMA 3. *$w\widehat{x}^2\dfrac{1-p(\widehat{x})}{p^2(\widehat{x})} + \widehat{v}(\widehat{x})$ is an unbiased estimator of* $\mathsf{Var}(\widehat{y})$.

4.3.3 Estimated Variance in Aggregation

Assume unbiased estimators $\{\widehat{x}_i\}$ of actual usages $\{x_i\}$, with associated unbiased variance estimators $\{\widehat{v}(x_i)\}$. When sampling is mutually independent for different i, the aggregate unbiased estimator $\sum_i \widehat{x}_i$ has variance whose unbiased estimator is $\sum_i \widehat{v}(x_i)$.

4.3.4 Example

To make the results of Sections 4.3.2 and 4.3.3 concrete, we describe the total variance estimator in the following composition of four sampling and aggregation operations: packet sampling, smart sampling of flow records, aggregation of flow records, and further smart sampling of the aggregated flow records.

- *Packet Sampling with Probability* $1/N$. Packets are sampled at a router, prior to formation of flow records. Consider flows i comprising packets with sizes $\{b_{ij}\}$ for $j = 1,\ldots,m_i$. The unbiased variance estimator is $\sum_i \sum_{j=1}^{m_i} u_{ij} b_{ij}^2 N(N-1)$ where u_{ij} is the selection indicator for packet j of flow i. Since individual packet sizes are not reported in flow record, we can usefully bound this expression above by $b_{\max}(N-1)\sum_i \widehat{x}_i$ where $\widehat{x}_i = N\sum_{j=1}^{m_i} u_{ij} b_{ij}$ is the unbiased estimator of usage in flow i.

- *Smart Sampling with Threshold z_1 of Raw Flows.* The flow records are smart sampled in the collector/aggregators. Each flow record i gives rise to a smart sampled usage estimate $\widehat{y}_i = v_i \max\{z_1, \widehat{x}_i\}$, where v_i is the indicator for selection of flow i. The variance associated with smart sampling $\widehat{x}_i$ is $v_i z_1 \max\{z_1 - \widehat{x}_i, 0\}$.

- *Aggregation of Smart Sampled Flows.* So far the cumulative sampling corresponds to the sampling operations of Theorem 1 with no report loss: $q = 1$.) At the collector, flow records matching keys in a given set are aggregated over time. Partition the set $\{1,\ldots,n\}$ of flows into disjoint sets I_k. Here k labels a key, and I_k is the set of flows with key k. Aggregate usage of flows with each key, yielding the usage estimates $\widehat{Y}_k = \sum_{i\in I_k} y_i$. The sampling variance for $\widehat{Y}_k$ is the sum of the sampling variance of each component $\widehat{y}_i$.

- *Smart Sampling with Threshold z_2 of Aggregated Flows.* The aggregates undergoing further smart sampling (with a different threshold) e.g. prior to long term storage. Each aggregated flow over key k gives rise to a smart sampled usage estimate $\widehat{S}_k = w_k \max\{z_2, \widehat{Y}_k\}$, where w_k is the indicator for selection of aggregate flow k. The variance associated with smart sampling $\widehat{Y}_k$ is $w_k z_2 \max\{z_2 - \widehat{Y}_k, 0\}$.

Combining these contributions, we arrive at the following unbiased estimate for the sampling variance:

$$\begin{aligned}\widehat{V} &= \sum_i \sum_{j=1}^{m_i} u_{ij} b_{ij}^2 N(N-1) + \sum_i v_i z_1 \max\{z_1 - \widehat{x}_i, 0\} \\ &\quad + \sum_k w_k z_2 \max\{z_2 - \widehat{Y}_k, 0\}. \qquad (7)\end{aligned}$$

Using the bound mentioned above for packet sampling, $\widehat{V}$ can be bounded above by the positively biased variance estimator

$$\begin{aligned}\widehat{V}' &= b_{\max}(N-1)\sum_i \widehat{x}_i + \sum_i v_i z_1 \max\{z_1 - \widehat{x}_i, 0\} \\ &\quad + \sum_k w_k z_2 \max\{z_2 - \widehat{Y}_k, 0\}. \qquad (8)\end{aligned}$$

The effect of the indicators v_i and w_k is to restrict the sums in which they occur to run over only terms that survive sampling. When used with actual data, each sum runs over only data points selected in sampling, with the accompanying indicators set to one. Thus, at each stage of sampling, the additional contribution to estimator variance can be computed from the information at hand.

5. PREDICTING THE PRODUCTION RATE OF FLOW RECORDS AT A ROUTER

In the next two sections of the paper we show how to predict the usage of TAP's resources from measured traffic statistics, and in particular its dependence on the sampling parameters. In this section we show how to predict the rate of production of flow records at the router. In Section 6 we show how to predict the rate of production of smart sampled flow records, and resources required to further aggregate them.

5.1 The Need to Model Bandwidth Usage

We argued in Section 4.2.3 that *any* amount of uniform (i.e. size independent) dropping of flow records is undesirable due to potential for large resulting variance in usage estimators. This motivates provisioning the collection bandwidth for flow reports to be sufficiently large to accommodate all reports. The rate of production of Sampled NetFlow records depends in a detailed way upon the composition of the network traffic that is being measured, and on the sampling parameters (the N of 1 in N sampling) and the flow interpacket timeout. Using a model for the distribution of packets within a flow, we estimate the mean number of flow records produced by sampling packets from a given original flow. Combined with either a model for distribution of original flow lengths and durations, or original flow records collected from actual traffic, we are able to estimate the rate of production of packet-sampled flow records

5.2 Modeling Packets within Flows

In order to determine the number of measured flows, we need a model for the splitting of sparse original flows. Consider a flow comprising n packets distributed over an interval of duration t. We adopt a model in which packets of the original flow are assumed to be independently and uniformly distributed in the interval of duration t. Packet are sampled independently with probability $1/N$; hence the number of sampled packets follows the binomial distribution $B_{1/N}(n)$, and the sampled packets are independently and uniformly distributed in the interval.

Note this model does suffer from some "edge effects" in that the first and last packets of the flow are not constrained to occur at the ends of the interval. Whereas it is possible to incorporate

this constraint in a more complex model, the resulting difference is small for increasing N since the first or last packets are selected with probability only $1/N$. Moreover, multiple measured flows will only occur when $n > N$, so again the details of placement of one packet will make little difference to subsequent results.

We note that another model for the distribution of packets has been considered in [8], namely that packets in the original flow are evenly spaced with the mean interpacket separation, and that packets are sampled periodically with random initial phase. In experiments we have found that the model of this paper is, in most cases, more accurate in predicting the average rate of production of measured flows.

5.3 Rate of Packet Sampled Flow Records

We now estimate the mean number of measured flows produced from an original flow under sampling. We take interpacket timeout as the only flow termination mechanism. We ignore the possibility of protocol-based termination, e.g. by observation of a TCP packet with the FIN flag set. On the other hand, only 1 in N of such packets will be sampled on average, so termination by observation of a FIN packet would be increasingly rare as N increases. We also ignore flow age as a criterion for termination. However, if the unsampled flows are measured flows, their ages do not exceed the allowed maximum. The same holds for a sampled flow, since its age cannot exceed that of the unsampled flow from which it is derived. Finally, we do not model termination for cache memory management.

THEOREM 2. *Let $f(n, t; N, T)$ denote the average number of measured flows produced from a single original flow comprising n packets randomly distributed over an interval of duration t, sampled independently with probability $1/N$, the measured flows having interpacket timeout T.*

$$f(n,t;N,T) = 1 + \left(\frac{\kappa - 1}{N} + 1\right)^{n-1} \left(\frac{\kappa(n-1)+1}{N} - 1\right), \tag{9}$$

where $\kappa = \max\{0, 1 - T/t\}$.

Theorem 2 can be used to estimates the rate of production of sampled NetFlow records in the two settings mentioned above:

- *Collected Unsampled Flow Records.* Here we estimate the average number of sampled flows that would be produced from a given set of unsampled flows. consider m flows collected over an interval of duration τ, flow i comprising n_i packets and having duration t_i. The total rate of sampled NetFlow records is estimated as

$$R = \tau^{-1} \sum_{i=1}^{m} f(n_i, t_i; N, T) \tag{10}$$

- *Modeled Distribution.* The results could also be used in conjunction with a model of flow length distributions. Let r be the arrival rate of original flows. Let $p(n, t)$ denote the model probability that a given flow comprises n packets distributed over a duration t. Then the total rate of sampled NetFlow records is estimated as

$$R = r \sum_{n,t} p(n,t) f(n,t;N,T) \tag{11}$$

Note that (11) can be regarded as arising from averaging (10) over a distribution of sample paths. However, since (10) is a sum over flows, it is not affected by correlations between flows, hence only the marginal distribution $p(n, t)$ enters.

In a separate study we have compared the predictions of (10) with values obtained from packet level traces subject to simulated packet sampling and flow formation. In case examined, estimation of the rate of packet sampled NetFlow records was accurate to within 10%, and often closer, over a wide range of sampling rates and flow interpacket timeouts.

5.4 Applications

We see two applications of the above estimates (10) and (11) for the mean rate of production of flow records:

- *Estimation from Unsampled Flows:* unsampled flow records are used to predict the rate at which packet sampled flow records would be produced. In this case, N is the sampling period for 1 in N packet sampling.
- *Estimation from Sampled Flows for Decreased Sampling Rate:* sampled flow records collected with 1 in M sampling are used to predict the rate of production were records to be collected with 1 in NM sampling for $N > 1$. In this case, N is the factor by which the sampling period is to be increased.

6. PREDICTING THE PRODUCTION RATE OF SMART SAMPLED FLOW RECORDS

In this section we show how to estimate the resources used by the smart sampled flow records at the collector. We focus on two cases. In the first, we estimate the output rate of flow records from smart sampling at the collector. This enables dimensioning of the storage and/or transmission resources required to accommodate the sampled records. In the second case, we consider further aggregation of the smart sampled flow records, and estimate the number of aggregate flows that result. In applications we expect aggregation to be performed over successive time windows. The estimates enable dimensioning of memory required for the aggregation table.

We perform these estimates in two ways. In Section 6.1 we derive an upper bound based on aggregate characteristics of the incoming stream of flow records. In Section 6.2 we obtain an estimate based on the detailed statistics of measured flows.

6.1 Smart Sampling Resources: Upper Bound

In the appendix we prove the following:

THEOREM 3. *Consider a stream of flow records arriving at average rate R, representing a data rate B. When this stream is smart sampled with threshold z, the expected rate R_s at which flows records are produced is bounded above as*

$$R_s \leq \min\{R, B/z\}. \tag{12}$$

Theorem 3 has two direct applications for the TAP architecture: the output load of the smart sampler, and storage resources for aggregation. In both cases B and z are the same: the data rate of the traffic being measured, and the sampling threshold respectively. The rate of production R of flow records from routers is to be determined from the methods of Section 5.

6.1.1 Bounding Output Rate of the Smart Sampler

R is the average rate at which flow records arrive at the smart sampler, R_s bounds the average rate of production of smart sampled flow records.

6.1.2 Bounding Resources for Aggregation

In TAP, the smart sampled raw flows are aggregated over a time interval τ (e.g. over one hour). The key used to aggregate may be

the just the raw flow key, or it may be coarser, e.g. a BGP routing prefix. We want to estimate the number of aggregate flows generated over the interval τ. Thus we want to determine the average rate $R_{\mathrm{s,agg}}$ at which unique keys (at the desired aggregation level) presented by flows that survive smart sampling during the period of length τ.

Clearly $R_{\mathrm{s,agg}}$ is bounded above by R_{s} (consider the case that all keys are unique). It must also be bounded above by the rate R_{agg}, the average rate, over the interval, at which unique aggregate keys become present in the NetFlow record prior to smart sampling. Since $R_{\mathrm{agg}} \leq R$,

$$R_{\mathrm{s,agg}} \leq \min\{R_{\mathrm{agg}}, B/z\}. \tag{13}$$

6.2 Smart Sampling Resources: Estimate

We now obtain a more detailed estimate that allows us to determine how tight the bound of Theorem 3 is. Ideally such an estimate would proceed by finding the distribution of the number and packet and byte lengths of the measured flows, then averaging the effect of smart sampling over this distribution. However, such an approach is computationally formidable; we opt instead for a simpler approach. Consider raw flows labeled by i having packet, duration and bytes (n_i, t_i, b_i), collected over a period of duration τ. Packet sampled NetFlow yields on average $f(n_i, t_i; N, T)$ measured flows. We apply these to the two examples for which bounds were obtained in Section 6.1.

6.2.1 Estimating Output Rate of the Smart Sampler

Assume that b represented bytes are allocated evenly amongst the average number $f_i = f(n_i, t_i; N, T)$ of flows. The expected number of smart-sampled flows that would arise from the original flow is $f_i p_z(b_i/f_i) = \min\{f, b_i/z\}$. Thus we estimate the rate of production of smart sampled flow records by

$$R_{\mathrm{s}} \approx \tau^{-1} \sum_i \min\{f(n_i, t_i; N, T), b_i/z\} \tag{14}$$

6.2.2 Estimating Resources for Aggregation

In TAP, the smart sampled raw flows are aggregated over a time interval τ (e.g. over one hour). The key used to aggregate may be the just the raw flow key, or it may be coarser, e.g. a BGP routing prefix. We want to estimate the number of aggregate flows generated over the interval τ. Thus we want to determine the number of unique keys (at the desired aggregation level) presented by flows that survive smart sampling during the period of length τ.

A given original flow i produced on average $f(n_i, t_i; N, T)$ measured flows with the same key. Suppose we can calculate the probability q_i that at least one of these flows survive smart sampling. Then the average rate at which aggregate flows are formed is

$$R_{\mathrm{s,agg}} = \tau^{-1} \sum_\kappa \left(1 - \prod_{i \in I_\kappa} (1 - q_i)\right) \tag{15}$$

where κ ranges over the set of aggregate flow keys present in the raw flows, and I_κ is the set of raw flows whose key matches κ.

Instead of calculating q_i exactly, we estimate it as follows. If $f_i < 1$ we treat it as the probability for exactly 1 measured flow to be submitted to smart sampling. With this interpretation, $q_i = f_i p_z(b/f_i)$. If $f_i \geq 1$ we treat is as a number of flows which are definitely submitted to smart sampling. If f_i were an integer, then we would have $q_i = 1 - (1 - p_z(b/f_i))^{f_i}$. We use the same expression for all $f_i \geq 1$. Combining, our estimate is

$$R_{\mathrm{s,agg}} \approx \tau^{-1} \sum_\kappa \left(1 - \prod_{i \in I_\kappa} (1 - q'_i)\right), \tag{16}$$

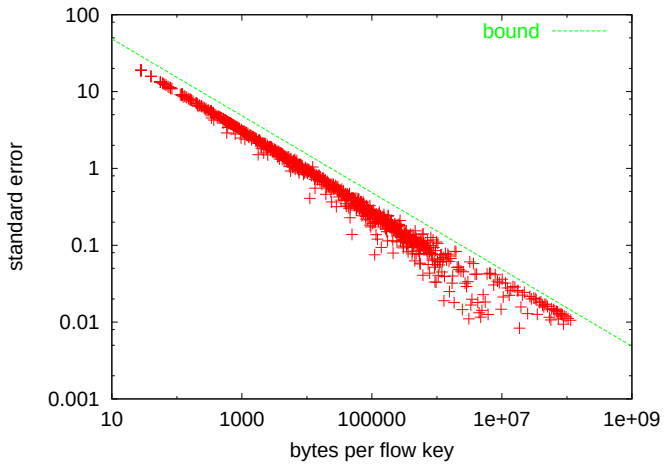

Figure 4: Variance and variance bound: for total bytes per flow, keyed by source address. Sampling period $N = 10$; smart sampling threshold $z = 10,000$; no transmission loss

where

$$q'_i = \begin{cases} f_i p_z(b/f_i) & f_i < 1 \\ (1 - p_z(b/f_i))^{f_i} & f_i \geq 1 \end{cases} \tag{17}$$

Note that for large z, q_i approaches b/z. Consequently, in this regime R_{agg} is approximately B/X *for any* aggregation scheme. Another way to see this is that since the q_i are small in this regime, $1 - \prod_{i \in I_\kappa}(1 - q_i) \approx \sum_{i \in I_\kappa} q_i$ and so the chance for at least one flow of a particular key to be selected is approximately the sum of the individual selection probabilities. In smart sampling, this will be their total represented bytes divided by z.

7. COMPARING ESTIMATES AND BOUNDS

In this section we compare the bounds and estimates on variance in Section 4 and resource consumption in Section 5 and 6, and show how to employ these results, using flow measurements taken from network traffic.

7.1 Description of Flow Trace

Our dataset comprised raw unsampled NetFlow records collected during a 1 hour on August 13, 2002. The data set recorded 2,019,840 raw flows containing 22,736,080,686 bytes distributed in 48,907,611 packets. The average data rate over the hour was thus 50.5 Mbits/sec.

7.2 Variance of Usage Estimates

We compare the simple bound of Theorem 1(iii) for the variance of usage estimates with the corresponding exact expression of Theorem 1(iii). We observed in Section 4 that it is highly preferable to use the bound on the grounds of computational and data simplicity. We now show that it provides a good approximation.

We performed the comparison over a joint range of packet sampling periods N from 1 to $10,000$ and smart sampling thresholds z from 1 to 10^9 bytes. The results shown in Figure 4, for $N = 10$ and $z = 10,000$ are typically. There is no transmission loss: $q = 1$. Keying flows by source IP address, we show a scatter plot of standard error $\sqrt{\mathsf{Var}\,\widehat{X}}/X$ against X for estimation of the total usage X in each color. Also shown is the bound of Theorem 1(iii). The bound is quite close. The greatest divergence between bound and variance typically occurs for larger X. Such usage X is more likely

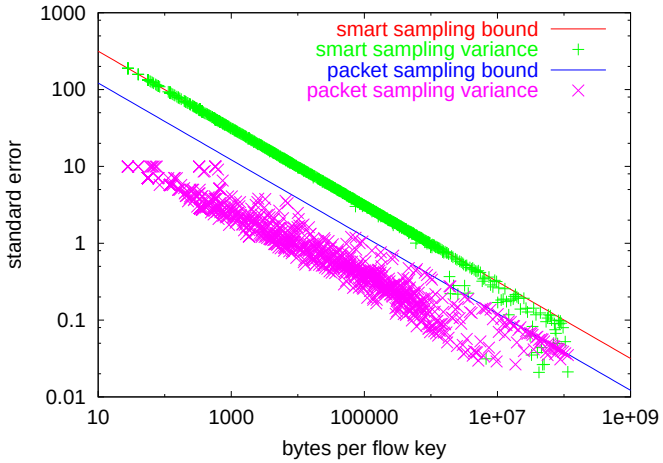

Figure 5: Variance and variance bound: for total bytes per flow, keyed by source address. Separate variance due to packet sampling with period $N = 100$; variance due to smart sampling with threshold $z = 1,000,000$; no transmission loss.

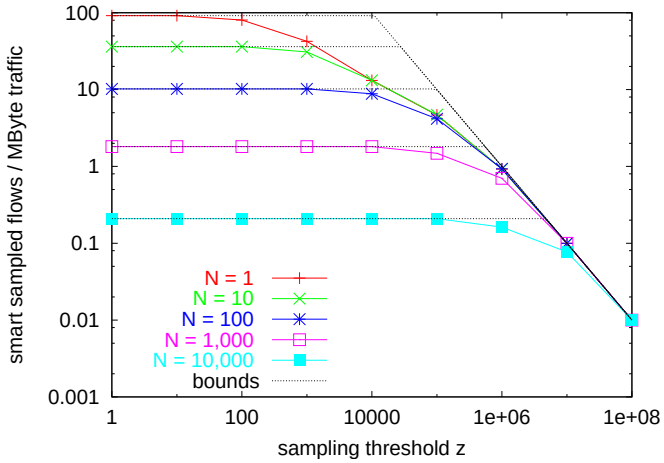

Figure 6: Smart sampled records: volume per MByte of original traffic, as function of smart sampling threshold for different packet sampling periods N. Also shown for each N: bound R_s from Theorem 1, where volume of packet sampled NetFlow records R is estimated by application of (10) to unsampled raw NetFlow records for each N.

to include component flows $x_i > z$ for which there will in fact be no sampling error.

We now show that individual components of the variance also have close upper bounds. We can identify in the variance of Theorem 1(ii) the component due to smart sampling (the first term) and that due to packet sampling (the last term). The corresponding standard errors can be compared with their respective bounds $\sqrt{z/X}$ and $\sqrt{(N-1)b_{\max}/X}$. We do this for $N = 100$ and $z = 10^6$ Bytes in Figure 5. Note that the bounds are individually tight: in each case there are variance points that lie on their associated bounds.

7.3 Application Volumes of Flow Records

A breakdown of traffic by application was conducted on the basis of well known application ports, as specified via RFC 3232[19]), and other identification made on the basis of specific application knowledge. 2,267 different such applications were identified. Six of the applications these each accounted for more than 1% of the byte total of the traffic; the percentage for each of these applications, along with the percentage bytes not attributable to an application, are displayed in Table 3. Observe that one p2p application constitutes nearly half the traffic volume.

		Flows per MByte, given N				
application	byte vol	1	10	10^2	10^3	10^4
kazaa	46.6%	30.29	12.18	5.67	1.03	0.12
gnutella	5.9%	68.08	36.20	11.21	2.08	0.23
napster	5.4%	12.65	11.24	8.69	1.95	0.22
www	2.9%	703.64	264.33	49.07	6.35	0.70
unidentified	2.6%	190.97	60.60	15.00	3.00	0.37
vrml-multi-use	1.2%	70.80	29.07	8.45	1.46	0.16
directx-gaming	1.1%	113.25	47.55	14.70	2.48	0.28

Table 3: Break down of percentage of byte volume, and by application, for applications accounting for at least 1% of total bytes. Predicted volume of raw flows, according to (10), as function of sampling period.

Using (10) we estimated the number of raw sampled NetFlow records that would be produced per Mbyte of original traffic, as a function of the sampling period. For the applications considered, clearly a considerable reduction in the rate of generation of NetFlow records is obtained through packet sampling. There is considerable variation in in the normalized rate of flow records amongst the applications: note the large rate for www traffic. This is attributable to the fact that users may be expected to run predominantly web clients rather than servers; their inbound traffic will comprise mainly http requests and ack packets for transfers. Web flows outbound to users are expected to have a lower rate of bytes per flow, while the packets per flow would be roughly the same. For the largest component, kazaa, we expect less asymmetry between inbound and outbound traffic: independent studies on similar links have found rough parity between inbound and outbound data volumes. However, it is not clear the extent to which this is due to individual users acting as bona fide peers, i.e. both downloading and serving content, as opposed to a balance in the aggregate behavior. These results underline the importance of understanding the application mix when estimate the likely volume of flow records.

7.4 Volume of Smart Sampled Flow Records

We compare the upper bound of Theorem 3 on the rate of production of smart sampled records with the corresponding estimate of (14). Figure 6 shows this comparison as a function of the sampling threshold z for the five cases of the sampling period for N from 1 to $10,000$. The bounds we obtained using the total byte rate B of the raw unsampled flow records, and applying (10) to determine the rate R of sampled raw flows for each N. Note that in an application, this number might be available directly from a collection of sampled NetFlow records. The curves were obtained by applying (14) to the raw unsampled flow statistics. For each curve the bounds essentially join projections of the initial and final portions. For large z the curves merge, since then most flows are subject to the same smart sampling $\widehat{x}_i < z$, and z play the role of the mean number of bytes represented per sampled flow.

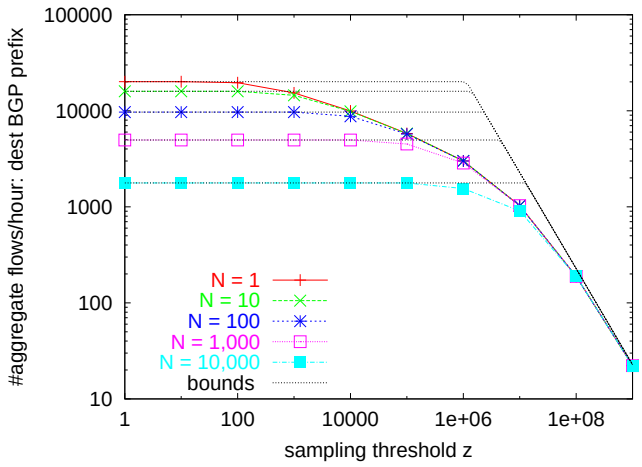

Figure 7: Aggregate flow: number of unique destination BGP prefixes per hour, as function of smart sampling threshold z, for different packet sampling periods N. Also shown for each N: bound R_s from Theorem 1, where rate R_{agg} of presentation of unique BGP prefixes is determined from NetFlow records.

7.5 Volume of Aggregated Smart Sampled Flows

We compare the upper bound (13) on the rate of production of aggregated smart sampled records with the corresponding estimate (16) obtained by modeling the smart sampling of individual flows. Figure 7 shows this comparison as a function of the sampling threshold z for the five cases of the sampling period for N from 1 to $10,000$. Flow keys are aggregated on destination BGP prefix. The bounds we obtained using the total byte rate B of the raw unsampled flow records; R_{agg} was determined by applying (16) to the raw unsampled flow records. In applications, R_{agg} may be available by collecting aggregate sampled NetFlow records directly. As with the unaggregated records, the curve merge for large z, which takes the role of the mean number of bytes represented per aggregated sampled flow. As remarked in Section 6.2.2, this property will hold for *any* aggregations scheme for large enough z.

8. CONCLUSIONS AND APPLICATIONS

This paper has described a Traffic Analysis Platform (TAP): a hierarchical infrastructure for the measurement and collection of traffic flow records. Sampling of packets and flows is required to manage consumption of TAP resources. This comes at the costs of introducing statistical uncertainty into traffic usage, since this must now be estimated from measurements. We gave a simple upper bound on the variance of usage estimates. This led us to the guideline, that the system should be run (under normal operational situations) in a way where flow loss is kept minimal and preferably avoided altogether. We gave simple bounds for the consumption of resources in the TAP architecture, and some estimates that make use of detailed flow records. These results constitute a set of tools that enable planning the resources of TAP infrastructure in order to meet accuracy goals in the estimation of network usage.

Further applications for smart sampling exist in the TAP architecture. Long-term archival of flow records requires potentially large amounts of storage. A further round of smart sampling can be used to reduce storage volumes while retaining the ability to recover detail from the archived data.

Appendix: Proofs of Theorems

Proof of Lemma 1: Using (6) $\mathsf{Var}(\hat{y}) = \mathsf{E}[\mathsf{Var}(\hat{y} \mid \hat{x})] + \mathsf{Var}(\mathsf{E}[\hat{y} \mid \hat{x}])$. Now $\hat{y} = w\hat{x}/p(\hat{x})$ where w is an indicator random variable that takes the value 1 with probability $p(\hat{x})$. Hence $\mathsf{Var}(\hat{y} \mid \hat{x}) = \hat{x}^2(1-p(\hat{x}))/p(\hat{x})$. (Note this is bounded at $x = 0$ due to the assumption of right continuity of $x/p(x)$ at $x = 0$.) Finally, $\mathsf{E}[\hat{y} \mid \hat{x}] = \hat{x}$, and the result follows.

Proof of Theorem 1: (i) $\mathsf{E}[\widehat{X} \mid \{\hat{x}_i\}] = \sum_{i=1}^n \hat{x}_i$ and $\mathsf{E}[\hat{x}_i] = x_i$. (ii) Applying Lemma 1,

$$\mathsf{Var}(\hat{y}_i) = \mathsf{E}[\hat{x}_i \max\{zq^{-1} - \hat{x}_i, 0\}] + \mathsf{Var}\,\hat{x}_i \le zq^{-1}x_i + \mathsf{Var}\,\hat{x}_i. \tag{18}$$

Similarly, conditioning on $\{u_{ij}\}$ and using (6) we find

$$\begin{aligned}
\mathsf{Var}\,\hat{x}_i &= \mathsf{E}[\mathsf{Var}(\hat{x}_i \mid \{u_{ij}\})] + \mathsf{Var}(\mathsf{E}[\hat{x}_i \mid \{u_{ij}\}]) && (19)\\
&= \mathsf{E}[\mathsf{Var}(Nq^{-1}\sum_{j=1}^{m_i} v_i u_{ij} b_j \mid \{u_{ij}\})] \\
&\quad + \mathsf{Var}(\mathsf{E}[Nq^{-1}\sum_{j=1}^{m_i} v_i u_{ij} b_j | \{u_{ij}\}]) && (20)\\
&= \frac{1-q}{q}\mathsf{E}[(N\sum_{j=1}^{m_i} u_{ij} b_j)^2] + \mathsf{Var}(N\sum_{j=1}^{m_i} u_{ij} b_j) && (21)
\end{aligned}$$

which yields the last two terms of (ii) after some algebra. The bound (iii) then follows easily.

(iv) We label by (i,k) the measured flows arising from the split of sampled packets from original flow i into ℓ_i measured flows labeled by k. The random variable r_{ijk} indicates the assignment of a sample packet to a measured flow: $r_{ijk} = 1$ if packet j from original flow i is sampled ($u_{ij} = 1$) and occurs in measured flow (i,k); otherwise $r_{ijk} = 0$. We will not need to specify a law for the $\{r_{ijk}\}$. Independent indicator variables v_{ik} take the value 1 if measured flow (i,k) is not dropped, this with probability q, and 0 otherwise. The usage estimate is $\widehat{X}' = \sum_{i=1}^n \hat{x}'_i$ where $\hat{x}'_i = \sum_{k=1}^{\ell_i} \hat{x}'_{ik}$, and $\hat{x}'_{ik} = Nq^{-1}\sum_{j=1}^{m_i} v_{ik}u_{ij}r_{ijk}b_j$.

Since $\sum_{k=1}^{m_1} r_{ijk} = 1$, $\hat{x}'_i$ is an unbiased estimator of x_i and hence $\widehat{X}'$ is an unbiased estimator of X. Repeating the decomposition (18), we find $\mathsf{Var}(\widehat{X}') \le zq^{-1}X + \sum_{i=1}^n \mathsf{Var}(\hat{x}'_i)$. Conditioning on $\{u_{ij}, r_{ijk}\}$ and using (6),

$$\mathsf{Var}(\hat{x}'_i) = \mathsf{E}[\mathsf{Var}(\hat{x}'_i \mid \{u_{ij}, r_{ijk}\})] + \mathsf{Var}(\mathsf{E}[\hat{x}'_i \mid \{u_{ij}, r_{ijk}\}]) \tag{22}$$

Now, by conditional independence of $\hat{x}'_{ik}$ given $\{u_{ij}, r_{ijk}\}$, the first term in (22) is

$$\begin{aligned}
\mathsf{E}[\mathsf{Var}(\hat{x}'_i \mid \{u_{ij}, r_{ijk}\})] &= \mathsf{E}[\sum_{k=1}^{\ell_i} \mathsf{Var}(\hat{x}'_{ik} \mid \{u_{ij}, r_{ijk}\})] \\
&= \frac{1-q}{q}\,\mathsf{E}[\sum_{k=1}^{\ell_i}(N\sum_{j=1}^{m_i} u_{ij}r_{ijk}b_{ij})^2] \\
&\le \frac{1-q}{q}\,\mathsf{E}[(N\sum_{k=1}^{\ell_i}\sum_{j=1}^{m_i} u_{ij}r_{ijk}b_{ij})^2] \\
&= \frac{1-q}{q}\,\mathsf{E}[(N\sum_{j=1}^{m_i} u_{ij}b_{ij})^2] && (23)
\end{aligned}$$

since $\sum_{k=1}^{\ell_i} r_{ijk} = 1$. For the same last reason, $\mathsf{E}[\hat{x}'_i \mid \{u_{ij}, r_{ijk}\}] = \mathsf{E}[\hat{x}_i \mid \{u_{ij}\}]$, and so combining with (19), (20), (22) and (23), we find $\mathsf{Var}(\hat{x}'_i) \le \mathsf{Var}(\hat{x}_i)$, and the result follows. □

Proof of Lemma 3: $\mathsf{E}[w\widehat{x}^2(1-p(x))/p^2(\widehat{x}) \mid \widehat{x}] = \widehat{x}^2(1-p(\widehat{x}))/p(\widehat{x})$ and so the result follows from Lemma 1. □

Proof of Theorem 2: Suppose there are $n \geq 2$ packet in the original flow and that $m \geq 2$ of these are selected. Let $\tau_1, \tau_2, \ldots, \tau_m$ be the (unordered) arrival times of the m packets. These times have joint probability distribution function PDF $g_m(\tau_1, \ldots, \tau_m) = t^{-m}$. Let the times between successive packets be $\sigma_1, \ldots, \sigma_{m-1}$. They have joint PDF h_m, where

$$\begin{aligned} h_m(s_1, \ldots, s_{m-1}) &= m! \int_{\tau_1 \leq (t - \sum_{i=1}^{m-1} s_i)} g_m(\tau_1, \ldots, \tau_m) d\tau_1 \ldots d\tau_m \\ &= m!(t - \sum_{i=1}^{m-1} s_i)/t^m, \end{aligned} \quad (24)$$

when $\sum_{i=1}^{m-1} s_i \leq t$, and 0 otherwise. The PDF h_m is invariant under permutation of the s_i, and hence all the σ_i have the same marginal distribution h_m^1, where

$$\begin{aligned} h_m^1(s_1) &= \int_{\sum_{i=2}^{m-1} s_i \leq t - s_1} h_m(s_1, \ldots, s_{m-1}) ds_2 \ldots ds_{m-1} \\ &= m(t-s_1)^{m-1} t^{-m}, \end{aligned} \quad (25)$$

for $s_1 \in [0, t]$, and 0 elsewhere.

The number of measured flows is 1 plus the number of s_i that exceed T. Hence the mean number of measured flows is $1 + (m-1)\mathsf{P}[\sigma_i > T]$. Note this expressions also holds for the case $m = 1$. For $T \geq t$, $\mathsf{P}[\sigma_i > T] = 0$. For $T \in [0, t)$, $\mathsf{P}[\sigma_i > T] = \int_t^T h_m^1(s_1) ds_1 = (1 - T/t)^m$. Combining these we find $\mathsf{P}[\sigma_i > T] = \kappa^m$ for all $T \geq 0$. Thus $f(n, t; N, T) = \sum_{m=1}^{n} \binom{n}{m} N^{-m}(1-1/N)^{n-m}(1 + (m-1)\kappa^m)$, which after some algebra is seen to be equal to (9). If $n = 1$ then $f = 1/N$, which agrees with (9) in this special case. □

Proof of Theorem 3: As before, consider total bytes X having an unbiased estimator $\widehat{X} = \sum_{i=1}^{n} \widehat{x}_i$ comprising a sum of n measured flows each of (random) size $\widehat{x}_i$, collected over an interval of duration τ. The expected average rate of smart sampled flows over the interval is

$$\begin{aligned} R_{\mathrm{s}} &= \tau^{-1} \sum_{i=1}^{n} \mathsf{E} p_z(\widehat{x}_i) =\leq \tau^{-1} \sum_{i=1}^{n} \min\{1, \mathsf{E}\widehat{x}_i/z\} \quad (26) \\ &= \tau^{-1} \min\{n, X/z\} = \min\{R, B/z\} \quad \square \quad (27) \end{aligned}$$

9. REFERENCES

[1] J. Apisdorf, K. Claffy, K. Thompson, and R. Wilder, "OC3MON: Flexible, Affordable, High Performance Statistics Collection," For further information see http://www.nlanr.net/NA/Oc3mon

[2] R. Cáceres, N.G. Duffield, A. Feldmann, J. Friedmann, A. Greenberg, R. Greer, T. Johnson, C. Kalmanek, B.Krishnamurthy, D. Lavelle, P.P. Mishra, K.K. Ramakrishnan, J. Rexford, F. True, and J.E. van der Merwe, "Measurement and Analysis of IP Network Usage and Behavior", IEEE Communications Magazine, vol. 38, no. 5, pp. 144–151, May 2000.

[3] B.-Y. Choi, J.Park, Zh.-L. Zhang, "Adaptive Random Sampling for Load Change Detection", ACM SIGMETRICS 2002 (Extended Abstract).

[4] Cisco NetFlow; for further information see http://www.cisco.com/warp/public/732/netflow/index.html

[5] K.C. Claffy, H.-W. Braun, and G.C. Polyzos. "Parameterizable methodology for internet traffic flow profiling", *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 8, . 1481–1494, October 1995.

[6] N.G. Duffield, C. Lund, M. Thorup, "Charging from sampled network usage," ACM SIGCOMM Internet Measurement Workshop 2001, San Francisco, CA, November 1-2, 2001.

[7] N.G. Duffield, C. Lund, M. Thorup, "Learn More, Sample Less: Control of Volume and Variance in Network Measurement", submitted for publication.

[8] N.G. Duffield, C. Lund, M. Thorup, "Properties and Prediction of Flow Statistics from Sampled Packet Streams," ACM SIGCOMM Internet Measurement Workshop 2002, Marseille, France, November 6-8, 2002.

[9] N. G. Duffield and M. Grossglauser, "Trajectory Sampling for Direct Traffic Observation", *IEEE/ACM Transactions on Networking*, vol. 9, pp. 280-292, 2001. Abridged version appeared in Proc. ACM Sigcomm 2000, Computer Communications Review, Vol 30, No 4, October 2000, pp. 271–282.

[10] C. Estan and G. Varghese, "New Directions in Traffic Measurement and Accounting", Proc SIGCOMM 2002, Pittsburgh, PA, August 19–23, 2002.

[11] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, F. True, "Deriving traffic demands for operational IP networks: methodology and experience", In Proc. ACM Sigcomm 2000, Computer Communications Review, Vol 30, No 4, October 2000, . 257–270.

[12] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, "NetScope: traffic engineering for IP networks", IEEE Network, vol. 14, no. 2, pp. 11–19, March-April 2000.

[13] A. Feldmann, J. Rexford, and R. Cáceres, "Efficient Policies for Carrying Web Traffic over Flow-Switched Networks," *IEEE/ACM Transactions on Networking*, vol. 6, no.6, pp. 673–685, December 1998.

[14] Inmon Corporation, "sFlow accuracy and billing", see: http://www.inmon.com/PDF/sFlowBilling.pdf

[15] D.G. Horvitz and D.J. Thompson, "A Generalization of Sampling without replacement from a Finite Universe", J. Amer. Statist. Assoc. Vol. 47, pp. 663-685, 1952.

[16] "Internet Protocol Flow Information " (IPFIX). IETF Working Group. See: http://net.doit.wisc.edu/ipfix/

[17] P. L'Ecuyer, "Efficient and portable combined random number generators", Communications of the ACM 31:742–749 and 774, 1988.

[18] J. Postel, "Transmission Control Protocol," RFC 793, September 1981.

[19] J. Reynolds, Ed., "Assigned Numbers: RFC 1700 is Replaced by an On-line Database", RFC 3232, January 2002.

[20] S.M. Ross, "Applied Probability Models with Optimization Applications, Dover, New York, 1970.