Properties and Prediction of Flow Statistics from Sampled Packet Streams

Nick Duffield Carsten Lund Mikkel Thorup
AT&T Labs–Research
180 Park Avenue, Florham Park, NJ 07932, USA
E-mail: {duffield,lund,mthorup}@research.att.com

Abstract— **Many routers can generate and export statistics on flows of packets that traverse them. Increasingly, high end routers form flow statistics from only a sampled packet stream in order to manage resource consumption involved.**

This paper addresses three questions. Firstly: what are the downstream consequences for the measurement infrastructure? Long traffic flows will be split up if the time between sampled packets exceeds the flow timeout. Using packet header traces we show that flows generated by increasingly prevalent peer-to-peer applications are vulnerable to this effect.

Secondly: can the volume of packet-sampled flow statistics be easily determined? We develop a simple model that predicts both the export rate of flow packet-sampled flow statistics and the number of active flows. It uses unsampled flow statistics—those commonly currently collected—as its data, i.e., it does not rely on having packet header traces available.

Thirdly: what properties of the original traffic stream can be inferred from the packet sampled flow statistics? We show that as well as estimating total bytes and packets, one can also infer more detail, specifically the number and average length of flows in the unsampled traffic stream, even though some flows will have no packets sampled. We believe that this information is useful, both for understanding source traffic, e.g. the dependence of flow lengths on application type, and also monitoring changes in the composition of the traffic, e.g., a flood of short flows during a DoS attack. In all cases, we evaluate our approach using packet header traces gathered in backbone and campus networks.

I. Introduction

A. Motivation

Many routers have the ability to generate and export statistics on flows of packets that traverse them. However, the consumption of computational and memory resources by the generation of flow statistics becomes onerous at line rate. For this reason, high-end routers increasingly use packet sampling to select a substream of packets, from which flow statistics are then formed.

This paper is motivated by the need to understand the consequences of packet-sampled flow formation for two broad questions. Firstly, we wish to determine resource usage, for both formation and transmission of flow statistics, and how it depends on the flow's characteristics (e.g. duration, size, source application) and the criteria used to identify flows for measurement. A spectrum of possible behavior can be demonstrated with packet traces reflecting today's Internet traffic; we wish to complement this approach with a model to determine resource usage for arbitrary traffic flows.

Secondly, we wish to estimate statistical properties of the original packet stream from the packet-sampled flow statistics. Original traffic volumes are relatively easy to estimate. However, to date there has been no work on recovering detailed properties of the original unsampled packet stream, such as the number and lengths of flows.

B. The Formation of Flow Statistics

An IP flow is a set of packets, that are 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 those IP header fields that are invariant along a packet's path.

A router keeps statistics on active flows passing through it. When a packet arrives at the router, the router determines if a flow is active for the packet's key. If not, is instantiates a new set of statistics for the packet's key. The statistics include counters for packets and bytes that are updated according to each packet that matches the key.

When the flow is terminated, its statistics are 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) [14] 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], [4], and are the subject of standardization efforts [11], [16]. Reported flow statistics 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. Examples of flow definitions employed as part of network management and accounting systems can be found in Cisco's NetFlow [3], Inmon's sFlow [10], Qosient's Argus [15], Riverstone's LFAP [17] and XACCT's Crane [19].

Flow statistics offer considerable compression of information over the header of the packet that comprise it, since the flow key is specified once for a given flow. In experiments, compression ratios of around 25 are not uncommon.

C. The Need for Packet Sampled Flow Statistics

The main resource constraint for the formation of flow statistics is at the router flow cache. To perform lookup of packet keys and counter increment at line rate would require the flow statistics to be stored in fast memory. However, core routers will carry increasingly large number of concurrent flows, necessitating large amount of fast memory: this would be expensive. By sampling the packet stream in advance of the construction of flow statistics, the time window available for flow cache lookup is prolonged, enabling storage to be carried out in slower, less expensive, memory.

D. Packet Sampling Methods

This paper considers sampling some target proportion p of the packet stream. There are several ways to implement this. In probabilistic sampling, the router makes a pseudorandom decision whether to sample each packet. In implementations, the decision could, for example, be governed by a pseudorandom number generator with well-known properties (see e.g. [12]) or be driven by the entropy of the packet contents itself (see e.g. [7]). When $p = 1/N$ for some integer N, periodic (or deterministic) sampling can be used, e.g. every N^{th} packet is selected.

Periodic sampling is very simple to implement: the router needs only decrement a counter. It has the potential disadvantage of introducing 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. However, 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 the selection of packets 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 necessary for flow formation in this case.

We mention some recent work in which the update of statistics of existing flow keys is performed only for a substream of packets [8]. This approach favors collection of statistics on longer flows. However, key lookup must still be performed for every packet.

E. Flow Semantics and Sampling

From the discussion of Section I-B it should be clear that an IP flow is an artifact of the manner in which the measuring device defines them, rather than having an independent existence. One motivation for the concept of a measured IP flow is that end hosts generate sets of packets as a result of transactions in applications, either automated or by users. A good definition of a flow, and in particular its starting and termination criteria, should encapsulate each transaction through the flow summary.

However, there are two factors that may hinder the effectiveness of such encapsulation. One is that new applications may generate packets in patterns that are not well captured by the flow definitions. The second factor, and most relevant for this paper, is that packet sampling removes cues for flow delineation from the packet stream. For example, termination of a TCP flow based on observation of a FIN packet is hindered if the packet is not present in the sampled stream. Thus interpacket timeout is expected to become the dominant method of termination for TCP flows when the sampling rate is low.

It may be advantageous to adjust flow delineation criteria with sampling rate in order to match the flow definition to the underlying nature of the transactions that generate the traffic. One case that we investigate in this paper is scaling the interpacket timeout inversely with the sampling rate in order to capture longer lived packet streams as a single flow.

The inherently artificial nature of flows provides something of a challenge for terminology. Our initial description of a flow as a set of packets with a common property observed in a given time interval serves to describe the flow independent of the measurement mechanism. We will sometimes use the term *original flow* to describe such a set of packets. Once a measurement mechanism—including termination criteria—has been defined, we can speak of a *measured flow*, together with the resulting flow statistic. Either type of flow can be called *sampled*; for an original flow this means a substream of packets sampled from it, while a sampled measured flow means a flow measured from such a substream.

F. Outline

This paper addresses three questions. Firstly: what are the downstream consequences for the measurement infrastructure? We call an original flow sparse if the typical interpacket time of the sampled packet stream exceeds the interpacket timeout for measured flows. A single sparse original flow gives rise to multiple flow statistics. The increasing prevalence of longer file transfers by peer-to-peer applications—as much as 50% of traffic on some links—may lead to sparseness if the sampling rate is sufficiently low. This observation motivates the first topic of our study. In Section II we use packet header traces to confirm that individual flows generated by streaming and peer-to-peer applications do generate multiple flow statistics at moderate sampling rates (e.g. 1 in 10 to 1 in 100).

Given sparseness, our second question is to ask whether the volumes of flow statistics, and the number of active flows, can be easily predicted. Packet traces are not available at most points in a network; heterogeneity of traffic prevents generalizing the analysis from a given trace to arbitrary network sites. Instead, in Section III, we develop a simple sampling model that works with more readily available data. Given a set of statistics of unsampled flows (perhaps derived directly from flow measurements) the model predicts the flow export rate and mean number of active flows that would result if we had instead formed flow statistics from a sampled version of the original packet stream. We evaluate our model using packet header traces.

Thirdly, we ask what properties of the original traffic stream can be inferred from the packet sampled flow statistics. Inference of total bytes and packets, possibly differentiated by flow key, is straightforward: dividing by the sampling rate the traffic rate represented in the measured flows yields an unbiased estimate of the original traffic rate. In Section IV we derive the variance of such estimates.

More difficult to infer are the detailed properties of the original flows: their arrival rate, their lengths. The main difficulty is that some flows may not be sampled at all; so it is not enough to simply form estimates through dividing the measured number of flows and their lengths by the sampling rate. When flow reports include supplementary protocol level information, specifically the occurrence of SYN flags within a TCP flow, we are able to form unbiased estimators of the flow rate and average lengths. We evaluate our method using packet header traces.

Although the method is confined to TCP traffic, this constitutes the overwhelming majority of Internet traffic. We also derive variances for these estimators, and find they are sufficiently accurate to monitor changes over timescales of a few seconds at high speed links. Thus, the method could potentially be used to detect changes in traffic characteristics, e.g., burst of short flows due to a SYN flooding attack. The method could also be used to characterize source traffic, e.g. mean flow lengths by application type. We conclude and propose further work in Section V.

G. Other Related Work

There has been some prior work on packet based sampling schemes and their consequences for estimation, although these works did not deal with the resulting flow statistics or estimation therefrom. Independent and deterministic 1 in N sampling, as well as stratified sampling out of finitely many bins, have been compared for packet sampling in [5]. The aim of this work was to efficiently estimate packet size distributions. Enabling inference from a sampled packet stream is one focus of the InMon's sFlow scheme; see [10]. A novel approach in this method is that packet reports are to include information of the size of the pool of objects from which packets are selected. This enables direct computation of the attained sampling rate. Use of the attained sampling rate for normalization of total byte and packet estimates can reduce estimator variance. Standards for network event sampling based on randomizing the inter-sample time have been set out in [13].

II. Impact of Sampling on Flow Statistics

A. Resource Usage and Sparseness

In this section we investigate the use of memory and transmission resources by packet-sampled flows. Memory usage will be characterized by the number of active flows. Transmission usage will be characterized by the rate at which flow statistics are exported, or equivalently, since we work in a fixed time window, the total number of flows exported. We shall use trace driven experiments to determine the variation of the number of active flows and export rate with sampling rate, broken down by application.

Consider an original flow with typical interpacket spacing τ. Suppose 1 in N packets are sampled from this stream. Typically, the interpacket spacing in the sampled stream is τN. If τN exceeds the flow interpacket timeout, then the original flow tends to decompose into a number of separate measured flows, depending on how the packets are bunched. The worst case is even spacing: each sampled packet would give rise to a separate measured flow. If $N\tau$ is less than the flow interpacket timeout, the reverse holds, and the packets tend to be reported as a single measured flow. If evenly spaced, a single measured flow would result; bunching may increase the number of measured flows.

This presupposes there are multiple packets in the sampled stream. We will term an original flow *sparse*, if

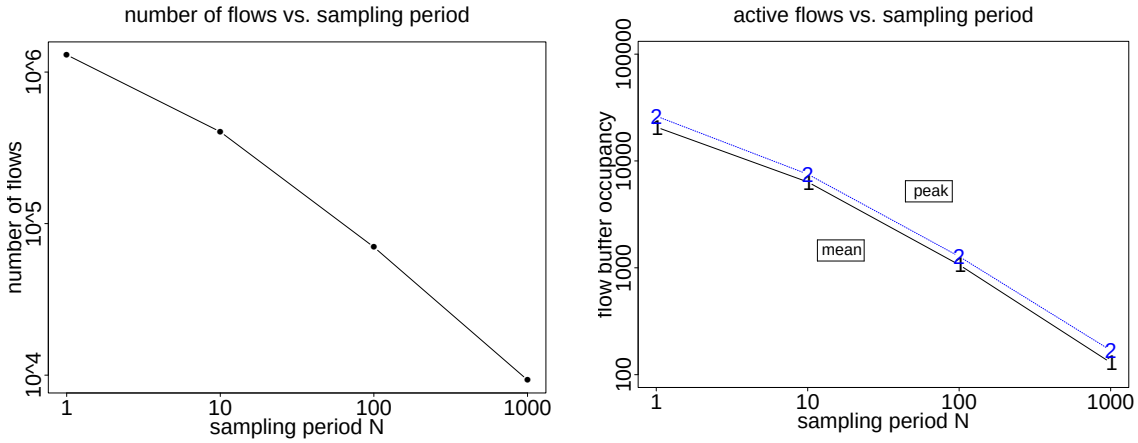

Fig. 1. Exported flows (left) and active flows (right, mean and peak) as function of sampling period N. Interpacket timeout for measured flows was 30 seconds.

sampling at a given rate typically yields more than one packet, with the typical interpacket time of the sampled packet exceeding the flow timeout. Thus an implicit and necessary condition for sparseness is that the typical flow length exceeds N. Packet size distributions of measured flows have previously been found to be heavy-tailed; see e.g. [9]. This leads us to expect a noticeable number of long, and hence potentially sparse, flows.

With sparse flows, packet sampling can then *increase* the number of measured flows and hence the downstream resource usage for those flows. Note we do not expect that sparseness to increase consumption of memory at the router. Whether or not splitting takes place, the original flow gives rise to at most one active measured flow at any time.

Peer-to-peer and streaming applications are candidates to produce sparse flows since they typically transmitted packets over extended periods. The recent rise in use of such applications is one driver for the study of the present paper. Following our terminology above we will call *sparse* those applications which may be expected to transmit sparse flows of traffic at typical sampling rates.

B. Experimental Traces and Flow Formation

An AT&T Labs PacketScope [2] was used to collect a packet header trace from a peering link, using the `tcpdump` tool. This preliminary study used a header trace comprising 10,000,000 IP packets collected over a period of 37 minutes starting Thursday April 26 19:35 2001 GMT.

By passing the header trace successively through two perl scripts, `sample` and `flows`, we determine the flows that would be produced after 1 in N sampling of the packet stream. `sample` accepts the header trace, one record per packet as input, and the sampling period N as a parameter, and outputs every N^{th} packet. `flows` takes a header trace as input, and a flow timeout t as a parameter. It constructs flows of packets with common source and destination IP addresses and TCP/UDP port numbers, with successive packets separated by no more than t. Packets using protocols other than TCP or UDP are discarded. Protocol specific information, such as TCP SYN or FIN packets, is not used demarcate flows. The output of `flows` comprises the constructed flow records, containing the address and port information, together with the total number of packets and bytes in the flow. Except for Section II-E, we use a flow timeout of 30 seconds.

C. Resource Usage by Aggregate Traffic

Figure 1(left) displays the number of exported flow statistics for deterministic sampling with periods N of 1, 10, 100 and 1,000. The dependence is roughly linear on the log scale, indicating the number of flows F behaves roughly as a power law $F \sim N^{\beta}$ for some power β. The slope of the final portion is -0.88. Model calculations show that, asymptotically for large N, the probability that any packet of a given flow is selected is asymptotically proportional to $1/N$ for large N, and thus we would expect the slope to approach -1 in this limit.

Figure 1(right) displays the average number of active flows over the duration of the trace. The dependence on N is similar to that of the flows. The peak number of active flows occupancy is only about 10% larger than the mean over the duration of the trace.

D. Variation by Application Type

Figure 2 breaks out the number of exported flows (left figure), and the mean number of active flows (right) by application, as determined from TCP/UDP port numbers. As the sampling period N increases, the number of flows

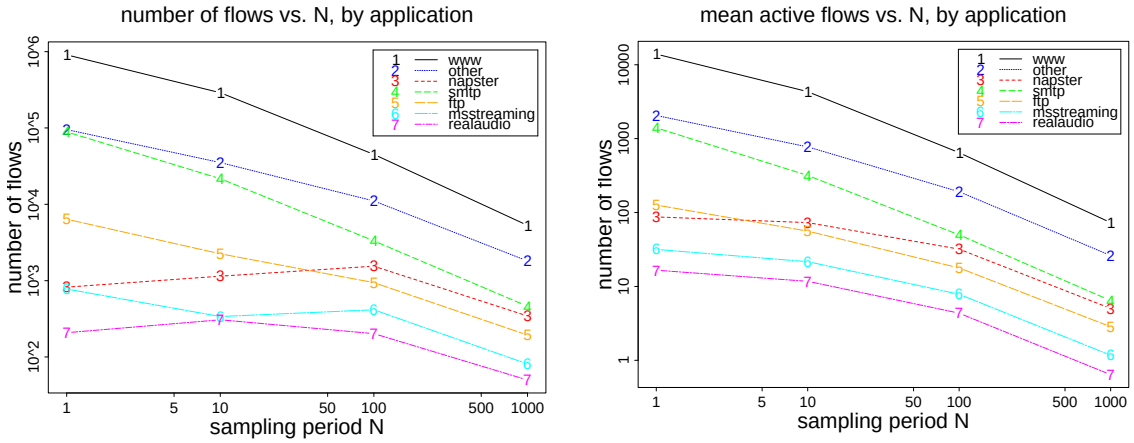

Fig. 2. Total flows (left) and active flows (right) for 30 second flow timeout, for selected applications. Sparse applications (napster, curve 3) and realaudio (curve 7) show initial *increase* in total flows with N.

	sampling period N			
appl.	1	10	100	1,000
www	6.12	1.91	1.23	1.04
other	22.42	6.06	1.93	1.15
napster	455.31	33.55	2.63	1.15
smtp	5.66	2.33	1.51	1.14
ftp	36.21	10.36	2.42	1.26
nntp	107.23	36.48	7.11	1.22
https	6.34	1.58	1.09	1.01
ms-strm	95.66	24.91	2.11	1.10
pop3	5.19	1.64	1.18	1.01
domain	2.26	1.34	1.07	1.01
realaudio	467.16	17.64	2.45	1.14
quake	19.58	15.30	5.14	1.25
http-alt	4.12	1.47	1.07	1.00
ALL	7.75	2.49	1.44	1.08

TABLE I
DEPENDENCE OF MEAN PACKETS PER FLOW ON SAMPLING PERIOD, BY APPLICATION; 30 SECOND INTERPACKET TIMEOUT

appl.	kbytes	prop.	kpkts	prop.
www	2,122,109	0.5202	5,500	0.5500
other	981,697	0.2406	2,140	0.2139
napster	282,356	0.0692	396	0.0396
smtp	211,280	0.0518	502	0.0502
ftp	126,433	0.0310	231	0.0231
nntp	101,214	0.0248	78	0.0078
https	96,671	0.0237	345	0.0345
ms-strm	41,903	0.0103	88	0.0088
pop3	33,537	0.0082	202	0.0202
domain	31,048	0.0076	204	0.0204
realaudio	26,783	0.0066	48	0.0048
quake	22,949	0.0056	261	0.0261
http-alt	1,762	0.0004	7	0.0007
TOTAL	4,079,741	1.0000	10,000	1.0000

TABLE II
PACKET AND BYTES: NUMBER AND PROPORTION, BY APPLICATION.

decreases for most applications. However, streaming and file sharing applications, in particular napster (curve 3) and realaudio (curve 7) buck this trend. These applications have the longest flow lengths (see Table I) and hence the most likely to rendered sparse. Observe that:

(i) Once the sampling period N reaches $N = 1,000$, the number of flows measured for sparse applications is no greater than for the unsampled packet stream (i.e. $N = 1$). This is because even for sparse applications, $N = 1,000$ exceeds the typical original flow length—see the $N = 1$ column of Table I—and so we expect that only 1 measured flow is produced from each original flow.

(ii) From Table I, once $N = 1,000$, the mean flow lengths are close to 1, and hence most flows have length equal 1. Sparse original flows have mostly been split into one-packet measured flows by sampling, while shorter original flows (i.e. of length less than N) will mostly have one packet sampled, if any.

(iii) in this trace, sparse applications constitute a relatively small proportion of the total bytes, being no more than about 10%: see Table II. However, use of peer to peer applications has risen since the trace used here was collected (April 2001). Flow statistics gathered at a large service provider show that up to 50% of traffic in some links can currently be attributed to peer to peer applications, such as Grokster, Gnutella, Kazaa and Morpheus.

(iv) for the sparse applications, the number of active flows—see Figure 2 (right)— decreases with N, although not as quickly as for other applications.

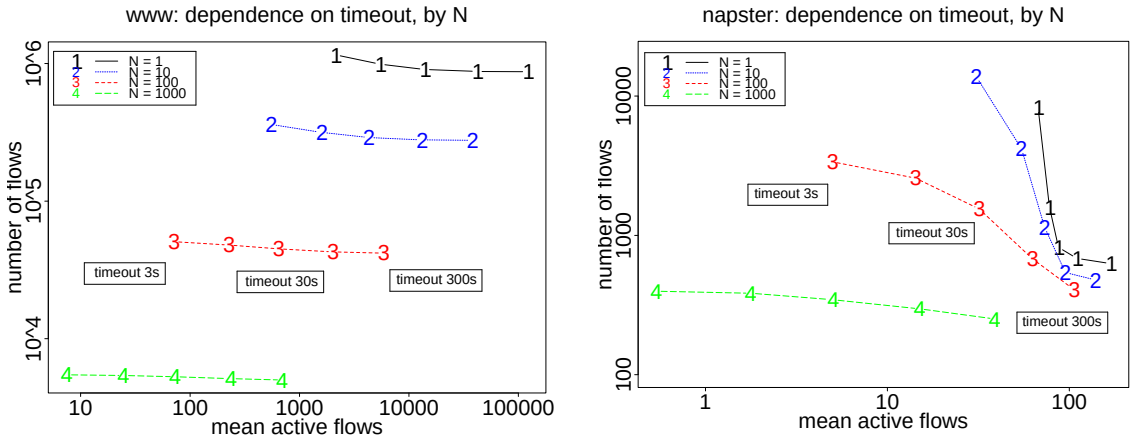

Fig. 3. Trade-off between total flows and active flows with interpacket timeout. (3, 10, 30, 100, 300 seconds). Trade-off for a sparse application (napster, right) more sensitive than for non-sparse application (www, left), when sampling period N smaller than typical flow length.

E. Trade-offs and Dependence on Flow Timeout

Lengthening the flow timeout could reduce the number of flows exported by sparse applications by grouping together packets that would otherwise be split into separate flows. However, this increases the number of active flows, since the flow cache entry must be kept open longer. Figure 3 shows the trade off between total flows and active flows, for a given sampling period N, as the flow timeout is varied.

For a non-sparse application, www, (Figure 3, left) there is little benefit in increasing the timeout: the active flows increase but the total flows decrease very little. From Table II, the mean original flow length is only about 6, so already for $N = 10$ there will be only 1 or a few packets reported on per original flow, whether in the same or different measured flows. Once N is large enough for 1 packet measured flows to be the norm, the total number of measured flows is roughly inversely proportional to $1/N$: the probability than a given packet is selected. This is borne out by the figure. Since the original flows were short, increasing the timeout increases the mean number of active flows roughly proportionately, but has little effect on the total number of flows, particularly for large N.

For a sparse application, napster, (Figure 3, right), the trade-off is more pronounced. From Table I, the mean original flow length, 455, is long enough that flows can be sparse for N as large as 100. However, for $N = 1,000$, the flows tend not to be sparse for any timeout, and the behavior is roughly the same as for non-sparse applications.

F. Summary

Sparse applications produce original flows comprising many packets with moderate interarrival times, that are likely to be split by packet sampling into multiple measured flows. In the trace examined, packet sampling can actually increases the number of measured flows exported for such applications. This phenomenon was observed for some peer-to-peer and streaming applications at moderate sampling rates: 1 in N for $N = 10$ and $N = 100$, but declines once $N = 1,000$, since this exceeds the typical original flow length.

The rise in use of peer to peer applications has been rapid and recent; other sparse applications may arise in future. This motivates the need to track flow length distributions and determine their effect, in conjunction with sampling rate choices, on the consumption of memory and transmission resource by measured flow statistics. In the next section, we use a simple sampling model that enables this to be done from unsampled flow statistics alone.

III. Predicting Resource Usage

In the previous section we used packet traces to investigate the dependence of memory and resource consumption by measured flows on application type, sampling rate and flow interpacket timeout. Although we accounted for the qualitative dependence, we have no reason to think the detailed behavior (e.g. traffic mix, mean flow lengths, the value of the power law in Figure 1) would be universal. However, in many cases packet header traces are not available to perform an equivalent analysis in a case of interest. In this section we assume that a set of unsampled measured flow statistics is available. A sampling model estimates the number of measured and active flows, had the original packet stream been sampled.

A. Flow and Packet Model

Consider an original flow comprising n packets distributed over a period of length t. Suppose packets are

sampled from this flow at mean rate $1/N$, and measured flows formed with timeout T from the samples. We wish to determine two quantities:

- $f(n,t;N,T)$: the resulting number of measured flows,
- $a(n,t;N,T)$: the total time the flow was active, i.e., the time elapsed between the first and last sampled packet, plus the timeout T.

Clearly, f and a are not determined without further assumptions concerning (i) the spacing of packets within the original flow, and (ii) the manner in which sampling is performed, e.g., deterministic or random. Since sampling involves random choice, the number of exported and active flows are themselves random quantities. Thus we must either provide their distribution, or summarize this through some statistic, such as their mean. To be specific, we determine the average values of f and a under deterministic sampling of an evenly spaced flow of packets. Whereas this model would only be accurate for sampling traffic comprising a single flow, we shall see it yields simple explicit results that are quite accurate in practice.

B. Deterministic Sampling with Equal Spacing

The packets of the original flow are assumed equally spaced in time. One in N of the packets are sampled periodically. The initial phase of the sampling is assumed random, i.e., packets $m, m+N, m+2N, \ldots$ are sampled, where m is distributed uniformly on $1, 2, \ldots, N$. We denote the expected value of f and a by f_1 and a_1.

Theorem 1: Assume deterministic sampling of a flow with period N and random initial phase.

$$f_1(n,t;N,T) = \begin{cases} 1, & Nt \le (n-1)T \text{ and } N < n \\ n/N, & \text{otherwise} \end{cases} \quad (1)$$

$$a_1(n,t;N,T) = \begin{cases} t\frac{n-N}{n-1} + T, & Nt \le (n-1)T \text{ and } N < n \\ nT/N, & \text{otherwise} \end{cases} \quad (2)$$

Proof: Assume first $n \le N$. At most 1 packet is selected, with probability n/N. If the measured flow exists, it comprises 1 packet, and hence had duration T, the timeout. Thus $f_1 = n/N$ and $a_1 = Tn/N$. Otherwise $n > N$. Then the separation between original packets is $t/(n-1)$ and hence the separation between any selected packets is $Nt/(n-1)$.

Suppose first $Nt/(n-1) \le T$. The separation of selected packets does not exceed the timeout, and selected packets form one flow: $f_1 = 1$. On average this flow has n/N packets, so it is active for $a_1 = t(n-N)/(n-1)+T$. On the other hand, if $Nt/(n-1) > T$, n/N packets are selected on average, each packet giving rise to one flow. The expected number of active flows is then $a_1 = Tn/N$. ■

Timeout	Sampling period N			
T	10	100	1,000	10,000
1	1.22	1.15	1.04	1.00
10	1.21	1.13	1.13	1.02
100	1.23	1.10	1.10	1.09
1,000	1.23	1.08	1.10	1.06

Timeout	Sampling period N			
T	10	100	1,000	10,000
1	1.18	1.08	1.01	1.00
10	1.21	1.13	1.08	1.01
100	1.23	1.11	1.10	1.05
1,000	1.23	1.09	1.10	1.05

TABLE III

ACCURACY OF FLOW PREDICTION: RATIOS PREDICTED TO ACTUAL NUMBER OF FLOWS. (UPPER) TOTAL FLOWS $\widehat{F}_1/F$; (LOWER) ACTIVE FLOWS $\widehat{A}_1/A$. FOR DIFFERENT TIMEOUTS T AND SAMPLING PERIODS N

C. Prediction and Accuracy

Given a set of original flows $i = 1, 2, \ldots, M$ of duration t_i and comprising n_i packets present during an interval of duration D, we estimate the total number of exported flows by $\widehat{F}_1$, and the mean number of active flows by $\widehat{A}_1$, where

$$\widehat{F}_1 = \sum_{i=1}^{M} f_1(n_i, t_i; N, T), \quad (3)$$

$$\widehat{A}_1 = D^{-1} \sum_{i=1}^{M} a_1(n_i, t_i; N, T) \quad (4)$$

We investigated the accuracy of the estimator $\widehat{F}_1$ by comparing it against the number of sampled flows constructed from packet header traces. For a single trace, Table III tabulates the ratios of $\widehat{F}_1$ and $\widehat{A}_1$ to their actual values, F and A, as a function of the sampling period N and timeout T. The same timeout T was used for the sampled and unsampled flows. Agreement is quite close: one expects estimates within about 10% of the true value for, say, $T = 30$ and $N = 100$. Note accuracy is better for larger N; as the sampling period exceeds the typical flow length, most original flows have only one packet sampled, and for these the detailed modeling of sparseness is irrelevant.

We also argue that for long sparse flows, f and a are reasonably insensitive to the detailed modeling of sparseness. Detailed examination of the traces reveals that for larger N and smaller T, the dominant contribution to F from sparse original flows comes from those flows which have many sampled packets (i.e. $n/N \gg 1$) and which are very sparse (i.e. $Nt/n \gg T$). Since for these flows,

Timeout	Sampling period N			
T	10	100	1,000	10,000
1	0.89	0.66	0.64	0.73
10	1.08	0.79	0.65	0.69
100	1.17	0.96	0.86	0.77
1,000	1.23	1.04	1.01	1.00

Timeout	Sampling period N			
T	10	100	1,000	10,000
1	1.40	4.24	13.32	57.38
10	1.31	2.23	7.10	14.46
100	1.43	1.99	2.53	2.82
1,000	1.25	1.19	1.24	1.18

TABLE IV
ACCURACY OF FLOW PREDICTION WITHOUT MODELING SPARSENESS: RATIOS PREDICTED TO ACTUAL NUMBER OF FLOWS. (UPPER) TOTAL FLOWS $\widehat{F}_2/F$; (LOWER) ACTIVE FLOWS $\widehat{A}_2/A$. FOR DIFFERENT TIMEOUTS T AND SAMPLING PERIODS N

the typical sampled interpacket time is much larger than the flow timeout, the sampled packets will tend to form single-packet measured flows, regardless of the detailed distribution of the sampled interpacket time.

D. Impact of Sparse Flows

Although we just argued a certain insensitivity to the details of sparseness, sparseness itself should not be ignored. This would amount to reducing the conditions in the upper lines of (1) and (2) to simply $N < n$, i.e., without testing whether the sampled interpacket time exceeds the measured flow timeout. We call the resulting functions f_2 and a_2, with corresponding estimates $\widehat{F}_2$ and $\widehat{A}_2$ of the total and mean active flows, analogous to (3).

We tabulate the ratios $\widehat{F}_2/F$ and $\widehat{A}_2/A$ in Table IV. Except for small N and large T, $\widehat{F}_2$ underestimates F, noticeably more than $\widehat{F}_1$ overestimates F. Overestimation of A by $\widehat{A}_2$ is more striking, particularly for large N and small T. By ignoring the sparseness, the measured flow is judged to be active over nearly the entire duration of the original flow, rather than for the subintervals of the original flows. Thus, ignoring sparseness can lead to substantial over-estimation of the mean number of active flows, and hence the buffering resources needed to accommodate them in a router.

IV. INFERENCE OF ORIGINAL FLOW CHARACTERISTICS

A. Goals and Impediments

In the third part of our work we turn the telescope of the previous section around: we ask what properties of the original packet stream can be determined from the measured flow statistics of the sampled packet stream. Clearly these are used by a variety of measurement based applications; they can also be of use in setting the parameters for sampling itself. We shall consider the following:

1. Bytes and packet counts per flow key. These form the raw data for usage-based applications, including traffic engineering, and usage-based accounting.
2. Number of original flows, differentiated by flow key or some aggregation thereof. Statistical properties of original flows are useful for characterizing source traffic.
3. Mean size of original flows. Given 1 and 2 above, we would already have an estimate at our disposal: packet count divided by number of flows. We shall discuss other possible choices. As well as helping source characterization, knowing the mean original flow size would also help in assessing resource usage by packet sampled flows, as detailed in Section III.

The goal is the find estimates for these characteristics and understand the accuracy of these estimates. Estimation of bytes and packets in the original flows is comparatively straightforward: one need only divide the corresponding totals in the measured flows by the sampling rate. We spell this out in Section IV-B, together with expressions for estimator accuracy.

By contrast, the number and length of original flows cannot be obtained through "multiplication by N", as shown by Table I for flow lengths. An impediment to estimating the original flow properties is that an unknown number of original flows will have *no* packets sampled and thus not show up in the measured flows at all. Note that this applies predominantly to short flows: those whose length is shorter that the inverse of the sampling rate. Then, most measured flows have length 1, and it will be difficult to distinguish different distributions of the original flows. Consider the following two cases:

(i) 1,000,000 original flows of size 2. Perform 1/10,000 packet sampling. The number of sampled flows of size 1 is a random number whose mean is 199.8, while the probability of producing at least one measured flow of size 2 is only about 1%.

(ii) 1,998,000 original flows of size 1. Performing 1/10,000 packet sampling the number of flows of size 1 is a random variable with mean 199.8, whose distribution is almost indistinguishable from the above case.

This demonstrates that a large difference in the number of original flows can be difficult to distinguish from the distributions of the measured flows.

One way to circumvent this problem is to use more information than the measured flow lengths alone.

(i) With a parameterized statistical model of the original flow size distribution, we could estimate parameters of the model from the measured flows. One difficulty here is arriving at a sufficiently rich model class. Although certain broad features, e,g, heavy tails, have been observed in experiments, no generally accepted model exists. We do not

pursue this approach in this paper.
(ii) Additional information may be provided in the flow records themselves. As an example, Cisco NetFlow reports the set of TCP flags that were set by any packet in the flow. In section IV-C we show how to use this information to estimate the number of original TCP flows, and hence their mean length. Although the method is confined to TCP traffic, this forms the overwhelming majority of current traffic in the Internet: roughly 95% of all traffic flowing in a set of high speed links in a major service provider.

B. Byte and packet totals

We start by setting out the relatively simple estimators for numbers of original bytes and packets. If packets are sampled at average rate of $1/N$, independent of their size, (for periodic sampling N would be an integer, but need not be in general) then estimates of the bytes and packet of the original traffic are obtained by multiplying the measured bytes and packets by N. This applies to the aggregate, or to subsets of flows of a given key.

To be specific, we can estimate the number P of original packets by the random variable $\widehat{P} = N\widehat{p}$ where $\widehat{p}$ is the total number of packets in the measured flows. Mathematically, we can write $\widehat{p} = \sum_{i=1}^{P} w_i$ where the w_i are random variables taking the value 1 (indicating that the packet was sampled) with probability $1/N$ and 0 (indicating that the packet was not sampled) with probability $1 - 1/N$. Each packet sampled contributes 1 to the sum, which is that the number of packets actually sampled.

$\widehat{P}$ is an unbiased estimator of P, i.e., its expectation equals P. In detail: $\mathsf{E}\widehat{P} = N\sum_{i=1}^{P} \mathsf{E}w_i = P$. Note independence of the w_i is not assumed.

For independent sampling, the w_i are independent, and we can write the variance $\mathsf{Var}\,\widehat{P} = PN^2\,\mathsf{Var}\,w_1 = PN(1 - N^{-1})$. The standard error $\sqrt{\mathsf{Var}\,\widehat{P}}/P$ is thus bounded above by $\sqrt{N/P}$.

Byte totals are estimated similarly, by the random variable $\widehat{B} = N\widehat{b}$ where $\widehat{b}$ is the total number of bytes in the measured flows. We write $\widehat{b} = \sum_{i=1}^{P} w_i b_i$ where the b_i are the byte sizes of the packets in the original flows. Each packet i sampled contributes b_i to the sum, which is hence the total bytes of sampled packets. With independent sampling, $\mathsf{Var}\,\widehat{B} = N(1 - N^{-1})\sum_{i=1}^{P} b_i^2$. Writing the average and maximum packet size as b_{av} and b_{max} respectively, we can bound the standard error of $\widehat{B}$ above by $\sqrt{N/P} \cdot b_{max}/b_{av}$. Summarizing:

Theorem 2: (i) Suppose a stream of P packets containing B bytes is sampled with average probability $1/N > 0$ independent of packet size, with reported packet and byte totals $\widehat{p}$ and $\widehat{b}$. Then

$$\widehat{P} = N\widehat{p}, \quad \text{and} \quad \widehat{B} = N\widehat{b} \tag{5}$$

are unbiased estimators of P and B.
(ii) The standard errors of $\widehat{P}$ and $\widehat{B}$ are bounded above as

$$\frac{\sqrt{\mathsf{Var}\,\widehat{P}}}{P} \le \sqrt{\frac{N}{P}}, \quad \frac{\sqrt{\mathsf{Var}\,\widehat{B}}}{B} \le \sqrt{\frac{N}{P}} \cdot \frac{b_{max}}{b_{av}}, \tag{6}$$

where b_{av} and b_{max} are the average and maximum packet size respectively.

C. Number of original TCP flows

The TCP protocol signals the start and end of connections with packets that are distinguished by flags (bits) in the code bits of the TCP header; see e.g. [6]. The first packet of a connection has a SYN flag set, whereas the last has the FIN flag set. NetFlow traces include the cumulative OR of the code bits. Thus by inspecting the code bits of the flow, we may determine whether or not the SYN and FIN flags were set on any packet detected in the flow. We will refer to a packet with a SYN flag set as a SYN packet. Here we assume:
(A1) original TCP flows start with a SYN packet
(A2) original TCP flows contain exactly one SYN packet.
Thus, if a SYN packet is reported in a measured flow, even in a sampled one, it must have been set in the first packet of the flow. We shall investigate the extent to which these assumptions are satisfied, and the effect of these violations, in Section IV-F.

A parallel methodology could be based on FIN flags, since all TCP sessions should end with a FIN packet. However, there may be many flows for which this is not the case: a SYN-flooding denial of service attack employs flows comprising one SYN packet.

C.1 Two estimators of TCP flow numbers

We now show that when packet sampling is used, the statistics of the flow code bits allow us to infer the number of original flows, M, that were present during the collection period. We will construct two estimators of M, each with distinct statistical advantages.

We assume that packets are selected with probability $1/N$. (At this point we do not assume independent selection of different packets). If the SYN packet is selected, then trivially the flow is sampled. Thus the number $\widehat{m}_1$ of measured flows that contain a SYN packet has expectation M/N. Consequently $\widehat{M}_1 = N\widehat{m}_1$ is an unbiased estimator of M.

One potential disadvantage of $\widehat{M}_1$ is that it uses only flows containing a SYN, i.e., only a subset proportion of the measured flows as its data. An alternative that addresses this issue is the following. We assume that no splitting of flows due to sparseness takes place; this can be realized through using an infinite flow timeout. We divide the original flows into three classes. S_1 comprises those flows for which the measured flow comprises exactly one

SYN packet. S_2 comprises those flows for which the measured flow has at least one non-SYN packet. Note that if any packet of a given original flow is sampled, then it must be in either S_1 or S_2. S_3 will denote the set of original flows from which no packet was sampled.

Assume now that sampling of the first packet of the original flow occurs independently of the other packets. Since $S_1 \subset S_2^c$ (here S_2^c denotes the compliment of S_2, i.e., $S_1 \cup S_3$), then for a given flow $\mathsf{P}[S_1] = \mathsf{P}[S_1 \mid S_2^c]\mathsf{P}[S_2^c] = (1 - P[S_2])/N$. Now let $\widehat{s}_1$ and $\widehat{s}_2$ denote the measured numbers flows stemming from S_1 and S_2 respectively. Define $\widehat{M}_2 = N\widehat{s}_1 + \widehat{s}_2$ and observe that $\mathsf{E}\widehat{M}_2 = MN\mathsf{P}[S_1] + M\mathsf{P}[S_2] = M$. Hence we have proved:

Theorem 3: Assume (A1) and (A2). Then $\widehat{M}_1$ is an unbiased estimator of M, and so is $\widehat{M}_2$ under the additional assumption of infinite flow interpacket timeout.

C.2 Estimator variance

We now compare the sampling variance of these estimators as the number of original flows M grows. Our model for this is the following. Consider a sequence of original flows of lengths $f_1, f_2, \ldots$. We assume that the f_i are i.i.d. random variables of finite mean $\overline{f}$. Consequently, by the Strong Law of Large Numbers ([18]) the average length $\overline{f}_M = M^{-1}\sum_{i=1}^M f_i$ of the first M flows converges almost surely to $\overline{f}$ as M grows. This is almost the only statistical behavior of the f_i that shall concern us. In what follows, we shall condition on a particular realization of the flow sizes, which hence appear fixed. The only statistical behavior comes from the sampling. The form of our results will depend on the flow sizes only through the mean $\overline{f}$, and hence will hold for almost all realization of the flow lengths.

Define indicator variables $(x_{ij})_{i=1,2,\ldots;j=1,\ldots,f_i}$ taking the value 1 if packet j of flow i is sampled, and 0 otherwise. Let $z_i = \prod_{j=2}^{f_i}(1 - x_{ij})$. No packets of flow i beyond the first are sampled if $z_i = 1$.

Theorem 4: Assume (A1) and (A2) and independent packet sampling with probability $1/N$.

(i) $M^{-1/2}(\widehat{M}_1 - M)$ converges in distribution to a Gaussian random variable of mean 0 and variance $\sigma_1^2 = N(1 - N^{-1})$.

(ii) Assume additionally an infinite flow interpacket timeout and that the flow lengths f_i are i.i.d. with finite mean. Then for almost all sequences of flow lengths, the conditional distribution of $M^{-1/2}(\widehat{M}_2 - M)$ converges to that of Gaussian random variable of mean 0 and variance $\sigma_2^2 = \sigma_1^2\overline{z}$ where $\overline{z} = \mathsf{E}[(1-N^{-1})^{f_1-1}]$ is the probability that no packets of a flow beyond the first get sampled.

Proof: (i) Write $\widehat{M}_1 = N\sum_{i=1}^M x_{i1}$. Since M_1 is a sum of i.i.d random variables and $\mathsf{Var}\, x_{ij} = (1 - N^{-1})/N$, $\mathsf{Var}\, M_1 = MN(1 - N^{-1})$. The result follows from the Central Limit Theorem; [18].

(ii) Flow i falls in S_1 if $x_{i1}z_i = 1$ and in S_2 if $z_i = 0$. Hence we can write $\widehat{M}_2 = \sum_{i=1}^M \{1 + (Nx_{i1} - 1)z_i\}$. Using (a) the independence of the terms for different flows i; (b) the formula $\mathsf{Var}\, AB = \mathsf{Var}\, A \cdot \mathsf{E}B^2 + \mathsf{Var}\, B \cdot (\mathsf{E}A)^2$ with $A = Nx_i - 1$; (c) $z_i^2 = z_i$; (d) $\mathsf{E}[z_i] = (1 - N^{-1})^{f_i-1}$; (e) $\mathsf{Var}\, x_{i1} = (1 - N^{-1})/N$, we find $\mathsf{Var}\, \widehat{M}_2 = MN(1 - N^{-1})\overline{z}_M$, where $\overline{z}_M = M^{-1}\sum_{i=1}^M (1 - N^{-1})^{f_i} \leq 1$. Each term in the last sum has common distribution with finite mean $\overline{z}$, and hence by the Strong Law of Large Numbers $\overline{z}_M$ converges almost surely to $\overline{z}$. The result then follows by the Central Limit Theorem. ■

We can restate Theorem 4 as saying that $\widehat{M}_1$ has standard error roughly $\sqrt{N/M}$ while $\widehat{M}_2$ has standard error roughly $\sqrt{\overline{z}N/M}$. Since $\overline{z} < 1$ (unless all flows comprise 1 packet) we have achieved the aim of reducing estimator variance. However, this is done at the potential cost of introducing bias. $\widehat{M}_2$ was formulated under the assumption that flows are not split. Flow splitting will lead to overcounting of flows in class S_2. For smaller samples is may be better to sustain such bias instead of suffering the increased variance, which is more noticeable for smaller samples. $\overline{z}$ is closer to 1 for larger N.

To give an example, consider geometrically distributed f_i. Then $\overline{z} = 1/(1 + (\overline{f} - 1)/N)$. If the mean flow length is much longer than the sampling period, $\widehat{M}_2$ will have noticeably lower variance than $\widehat{M}_1$. Other the other hand, these are the conditions, depending on the flow timeout, under which sparse flows could be split up into several measured flows. The difference $\widehat{M}_2 - \widehat{M}_1$ actually gives us some measure of this, since it can be rewritten as the number of flows with property S_2 minus N times the number of flows with property S_2 that also contain a SYN: $\widehat{M}_2 - \widehat{M}_1$ estimates the additional flows that arise due to splitting.

D. Mean Length of Original TCP Flows

Equipped with estimates of the number of packets (Section IV-B) and the number of flows (Section IV-C) we can straightforwardly estimate the mean length of TCP flows as $\widehat{f} = \widehat{P}/\widehat{M}_k$ for either $k = 1$ or 2. Although these are not unbiased estimators of $\overline{f}$, they satisfy the following property:

Theorem 5: Under the assumptions of Theorem 4(ii), each $\widehat{P}/\widehat{M}_k$ is, for almost all sequences of flow lengths, a consistent estimator of $\overline{f}$, i.e., they converge to $\overline{f}$ almost surely as M grows.

Proof: Write $\widehat{P} = N\sum_{i=1}^M \sum_{j=1}^{f_i} x_{ij}$. By assumption, $\widehat{P}/M$ and $\widehat{M}_k/M$ are sums of independent random

variables with finite means $\overline{f}$ and 1 respectively. The result then follows by the Strong Law of Large Numbers. ■

Theorem 6: Under the assumptions of Theorem 4(ii), for $k = 1, 2$ each $M^{1/2}(\widehat{P}/\widehat{M}_k - \overline{f})$ is asymptotically normally distributed with mean zero and variances η_k^2 as follows:

$$\eta_1^2 = N(1 - N^{-1})\overline{f}(\overline{f} - 1), \quad (7)$$
$$\eta_2^2 = N(1 - N^{-1})\left(\overline{f}(\overline{f} - 1)\overline{z} + \overline{f}(1 - \overline{z})\right) \quad (8)$$

Proof: The variance is estimated by using the δ-method; see [18]. This supposes a sequence $M^{1/2}(\widehat{X}_1^M, \ldots, \widehat{X}_m^M)$ of vector valued random variables that is asymptotically Gaussian, as M grows, with mean 0 and asymptotic covariance matrix $(c_{ij})_{i,j=1,\ldots m}$. If g is a real function on $\mathbb{R}^m$ differentiable about 0, then $M^{1/2}(g(\widehat{X}_1, \ldots, \widehat{X}_m) - g(0))$ is asymptotically Gaussian, as M grows, with mean 0 and asymptotic variance $g'(0) \cdot cg'(0)$. (Here g' is the derivative of g). We express $\widehat{f}$ as $g(\widehat{P}/M - \overline{f}, \widehat{M}_k/M - 1)$ where $g(x, y) = (x + \overline{f})/(y + 1)$. Then $g'(0) = (1, -\overline{f})$ and it remains only to calculate the covariance matrix c of the numerator and denominator terms in $\widehat{f}$.

c_{11} is the same for both estimators, namely $\lim_{M\to\infty} M^{-1} \sum_{i=1}^{M} \mathsf{Var}(N \sum_{j=1}^{f_i} x_{ij}) = N(1 - N^{-1})\overline{f}$. For $\widehat{M}_1$, $c_{12} = \lim_{M\to\infty} M^{-1} \sum_{i=1}^{M} \mathsf{Cov}(N \sum_{j=1}^{f_i} x_{ij}, Nx_i) = N(1 - N^{-1})$, while $c_{22} = \lim_{M\to\infty} M^{-1} \mathsf{Var}\, \widehat{M}_1 = N(1 - N^{-1})$ from Theorem 4. Using $\widehat{M}_2$, $c_{12} = \lim_{M\to\infty} M^{-1} \sum_{i=1}^{M} \mathsf{Cov}(N \sum_{j=1}^{f_i} x_{ij}, (Nx_{i1} - 1)z_i) = N(1 - N^{-1})\overline{z}$, while $c_{22} = \lim_{M\to\infty} M^{-1} \mathsf{Var}\, \widehat{M}_2 = N(1 - N^{-1})\overline{z}$ from Theorem 4. The stated results then follow from these forms for c. ■

E. Timescales and Accuracy

We can interpret Theorem 6 as saying that the standard deviation of $\widehat{P}/\widehat{M}_1$ is roughly $\overline{f}\sqrt{N/M}$. Substituting $\widehat{M}_1 = N\widehat{m}_1$ for M, we estimate the likely error due to sampling as $\widehat{\eta}_1 \approx \overline{f}/\sqrt{\widehat{m}_1}$.

We can re-express this a saying that the relative error of estimation of the flow length is $\eta_1/\overline{f} \approx 1/\sqrt{\widehat{m}_1}$. Consider flows collected over a period of duration t from a link carrying traffic at rate C bytes/second. Assume a mean packet length p bytes, and ballpark mean flow length of ℓ. Then with sampling rate $1/N$ we expect $\widehat{m}_1 \approx Ct/(Np\ell)$. Taking a full OC48 (2.4Gb/second) and supposing $N = 100$, $p = 500$ and $\ell = 20$ we obtain a relative error of about $0.06/\sqrt{t}$. This shows that tracking flow lengths each second would have a relative error of only about 6%, probably sufficient to detect changes in the composition of aggregate traffic whose manifestations included a change in flow lengths. One example we have in mind is a burst in proportion of short flows due flooding with SYN packets that occurs in some denial of service attacks.

trace	packets	flows	SYN	FIN
full	9,566,657	354,950	315,200	283,357
reduced	6,889,444	299,875	315,067	270,081

TABLE V
TRACE PROPERTIES OF FULL TRACE AND REDUCED SUBTRACE OF FLOWS STARTING WITH SYN PACKET. FLOWS TERMINATED BY 30 SECOND TIMEOUT.

F. Evaluation of TCP Flow Length Estimators

We evaluate the method on a trace of TCP packets, both a raw version, and a reduced version constructed to better conform to the assumptions. Even for the raw trace, we find that estimation is accurate to within 10% at a sampling rate of 1 in 1,000.

F.1 Packet Trace

We evaluated the estimators $\widehat{P}/\widehat{M}_k$ of mean flow length using a second trace of 9,566,657 TCP packets collected over a period of 5 hours near the boundary of a corporate campus network. Note that SYN flagged packets for a given flow may be missing from the trace, due to flows having started before the trace was initiated, collection errors when taking the packet header trace, or network packet drops. In order to evaluate our approach under conditions that more closely matched the assumptions for which it was formulated, we constructed a subtrace comprising only those packets belonging to flows—as delineated by a 30 second timeout—that started with a SYN packet. This resulted in a a trace of 6,889,444 packets distributed in 299,875 such flows.

Properties of the full and reduced trace are shown in Table V. This shows that constructing the subtrace discarded about 28% of the packets comprising 16% of the flows, while the discarded flows contain less than 0.05% of the SYN packets in their interior. On the other hand, 5% of the total SYN packets in the reduced trace did not start a flow. These figures suggest discarded packets came from longer original flows, either ones that were already in progress (i.e. the initial SYN occurred before the trace was initiated) or flows that were already sparse with a 30 second interpacket timeout. In the reduced trace, SYN could occur in flow interiors due to retransmission if loss occurred downstream in the packets path from the collector, or when a second TCP connection followed one with the same key before the interpacket timeout expired. One could guard against the latter by terminating flows on occurrence of a FIN packet; however the table shows that

N	$\widehat{P}/\widehat{M}_1$	$\widehat{P}/\widehat{M}_2$	$\widehat{P}/\widehat{M}_2(S)$	$\widehat{m}_1$	$\widehat{\eta}_1$
1	22.97	22.97	22.97	299875	n/a
10	22.39	22.11	22.49	30767	0.12
100	22.48	21.88	22.25	3064	0.40
1,000	22.00	21.69	21.84	313	1.23

N	$\widehat{P}/\widehat{M}_1$	$\widehat{P}/\widehat{M}_2$	$\widehat{P}/\widehat{M}_2(S)$	$\widehat{m}_1$	$\widehat{\eta}_1$
1	31.90	27.11	27.11	299875	n/a
10	30.75	29.34	30.80	31116	0.17
100	30.63	29.27	30.00	3123	0.55
1,000	29.52	29.27	29.46	313	1.67

TABLE VI

ESTIMATED MEAN FLOW LENGTHS: REDUCED TRACE (UPPER) AND FULL TRACE (LOWER). $\widehat{P}/\widehat{M}_1$, $\widehat{P}/\widehat{M}_2$ AND $\widehat{P}/\widehat{M}_2(S)$ WITH FLOW TIMEOUT SCALING. MEASURED NUMBERS $\widehat{m}_1$ OF FLOWS CONTAINING SYN PACKETS, AND ESTIMATOR $\widehat{\eta}_1$ OF STANDARD DEVIATIONS.

10% of the flows did not terminate with a FIN, more than did not start with a SYN.

F.2 Accuracy: Mean Flow Length

The predicted flow lengths for sampling periods $N = 1, 10, 100$ and $1,000$ ranging from 1 to $1,000$ are shown for the reduced trace in the upper table of Table VI. The true mean flow length for this trace is the $N = 1$ entry of the column for $\widehat{P}/\widehat{M}_1$. Observe that $\widehat{P}/\widehat{M}_1$ is uniformly closer to the true value than $\widehat{P}/\widehat{M}_2$. We believe this is due to the bias in $\widehat{M}_2$ in overcounting flows discussed previously. We also tabulate $\widehat{m}_1$, the number of flows with a SYN, and $\widehat{\eta}_1$, the estimator of the standard deviation η_1 of $\widehat{P}/\widehat{M}_1$. If our assumption (A2) were obeyed, we would expect $\widehat{m}_1$ to decrease inversely with N. However, the decrease is a little slower than this, a symptom of the fact that the trace contains SYN packets that do not start flows, and hence $\widehat{M}_1$ overcounts of the number of flows. Indeed, we would expect $\widehat{M}_1$ to be close to the number Y of SYN packets if N and Y/N are large; this is borne out by Tables V and VI.

One way to alleviate overcounting in $\widehat{M}_2$ due to splitting of sparse flows is to lengthen the flow timeout; for example making it proportional to N. The resulting estimate is shown under the column $\widehat{P}/\widehat{M}_2(S)$. Although this offsets the bias somewhat, $\widehat{P}/\widehat{M}_1$ is, mostly, more accurate. Actually, the estimate $\widehat{P}/\widehat{M}_1$ is rendered slightly more accurate by the same method.

We can compare the difference between $\widehat{P}/\widehat{M}_1$ and actual mean flow length with $\widehat{\eta}_1$. For $N = 10$, the difference is about $7\widehat{\eta}_1$, greater than one would expect statistically, and hence indicative of bias. But at $N = 1,000$ the difference is only about $\widehat{\eta}_1$, so indistinguishable from usual statistical variation.

In the lower table we report the same results for the *full* trace. The relative accuracies of the estimators is unchanged. Deviations from the true values are greater than for the reduced traces, and greater than would be expected from sampling variation alone, they are still quite accurate: within 10% of the true value even at $N = 1,000$.

V. CONCLUSIONS AND FURTHER WORK

In this paper we have examined the consequences of collecting packet sampled flow statistics. We pointed out the flows in the original stream whose length is greater than the sampling period tend to give rise to multiple flow reports when the interpacket time in the sampled stream exceeds the flow timeout. In practice this occurs predominantly for traffic generated by peer-to-peer applications. Such traffic is on the rise, motivating the need to better understand the implications for resource usage in the measurement infrastructure of such splitting. To this end, we developed a model to predict sampled flow export rates and the number of active flows from original traffic statistics. Using traffic traces, we found the model to predictions accurate to within about 10% of number of total flows produce in a period and the mean number of active flows. Failing to take account of sparse flows (those vulnerable to splitting) can lead to underestimation of the total flows, and severe overestimation of the size of the buffer needed to accommodate active flows. We argued that the predictions would be relatively insensitive to the model details. Work in progress is to substantiate this assertion for other packet sampling models.

Turning the problem around, we also showed how to infer characteristics of the original traffic flows from the measured packet sampled flows. Whereas byte and packet volumes are estimated simply by dividing the measured quantities by the sampling rate, this approach does not work to estimate the number and mean length of flows, since some original flows will not be sampled at all. Instead, we introduced a method that exploited the statistics of reported SYN packets for TCP flows. In work in progress, we generalize our method to infer the distribution of the lengths of original flows, not just the mean. Further work will determine the effectiveness of the method in understanding the characteristics of substreams of the traffic, e.g. according to application.

Acknowledgments

We thank Cristian Estan and Matt Grossglauser for useful discussions.

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] Cisco NetFlow; for further information see http://www.cisco.com/warp/public/732/netflow/index.html and http://www.cisco.com/univercd/cc/td/doc/product/software /ios120/120newft/120limit/120s/120s11/12s_sanf.htm

[4] 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, pp. 1481–1494, October 1995.

[5] K.C. Claffy, G.C. Polyzos, and H.-W. Braun. "Application of Sampling Methodologies to Network Traffic Characterization", *Computer Communication Review*, 23(4):194–203, October 1993, appeared in Proceedings ACM SIGCOMM'93, San Francisco, CA, September pp. 13–17, 1993.

[6] D. Comer, "Internetworking with TCP/IP, Volume 1: Principles, Protocols, and Architecture", Third Edition, Prentice Hall, NJ, 1995.

[7] N. G. Duffield and M. Grossglauser, "Trajectory Sampling for Direct Traffic Observation", *IEEE/ACM Transactions on Networking*, v. 9 no. 3 (June 2001) pp. 280-292. Abridged version appeared in Proc. ACM Sigcomm 2000, *Computer Communications Review*, vol. 30, no. 4, October 2000, pp. 271–282.

[8] C. Estan and G. Varghese, "New Directions in Traffic Measurement and Accounting", ACM SIGCOMM Internet Measurement Workshop 2001, San Francisco, CA, November 1-2, 2001.

[9] 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.

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

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

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

[13] V. Paxson, G. Almes, J. Mahdavi, M. Mathis, "Framework for IP Performance Metrics", RFC 2330, available from: ftp://ftp.isi.edu/in-notes/rfc2330.txt, May 1998.

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

[15] Qosient, "Argus": http://www.qosient.com/argus/index.htm

[16] Real Time Flow Measurement, see: http://www.auckland.ac.nz/net/Internet/rtfm/.

[17] Riverstone Networks, Inc., see: http://www.riverstonenet.com/technology/

[18] M.J. Schervish, "Theory of Statistics", Springer, New York, 1995.

[19] XACCT Technologies, Inc., see: http://www.xacct.com