A methodology for studying persistency aspects of Internet flows

Jörg Wallerich⋆, Holger Dreger⋆, Anja Feldmann⋆, Balachander Krishnamurthy⋄, Walter Willinger⋄

⋆Technische Universität München {hdreger,feldmann,jw}@net.in.tum.de *

⋄AT&T Labs Research,Florham Park, NJ, USA {bala,walter}@research.att.com

ABSTRACT

We focus in this paper on Internet flows, consider their contributions to the overall traffic per time unit or bin, and perform a multi-scale and multi-protocol analysis to explore the persistency properties of those flows that contribute the most (also known as "heavy hitters" or "elephants"). Knowing the persistency features (or a lack thereof) of the heavy hitters and understanding their underlying causes is crucial when developing traffic engineering tools that focus primarily on optimizing system performance for elephant flows.

The main difficulty when studying the persistency properties of flows is that the available measurements are either too fine-grained to perform large-scale studies (i.e., packet-level traces) or too coarse-grained to extract the detailed information necessary for the purpose at hand (i.e., Netflow traces, SNMP). We deal with this problem by assuming that flows have constant throughput through their lifetime. We then check the validity of this assumption by comparing our Netflow-derived findings against those obtained from directly studying the corresponding detailed packet-level traces.

By considering different time aggregations (e.g., bin sizes between 1–10 minutes) and flow abstractions (e.g., raw IP flows, prefix flows), varying the definition of what constitutes an "elephant", and slicing by different protocols and applications, we present a methodology for studying persistency aspects exhibited by Internet flows. For example, we find that raw IP flows that are elephant flows for at least once (i.e., one bin or time unit) in their lifetimes tend to show a remarkable persistence to be elephants for much of their lifetimes, but certain aggregate flows exhibit more intricate persistency properties.

Keywords: Zipf's law, large flows, time-scales, flow-abstraction, elephants vs. mice, reservations, protocol use, Netflow

1. INTRODUCTION

This paper contributes to the nascent literature on characterizing the rates at which IP flows transmit data in the Internet [1, 2, 3]. In particular, we present the findings of an empirical study that demonstrate that Zipf's law for flow rates (i.e., number of bytes transmitted by the different flows during a given bin) holds not only for a fixed bin or time period, but applies more generally bin-by-bin[1] over time and for different flow abstractions. Recall that Zipf's law (see [4, 5] and references therein) for flow rates states that for a set of n inferred flow rates, ordered as $x_{(1)} \geq x_{(2)} \geq \ldots \geq x_{(n)}$, so we may think of $x_{(r)}$ as the rth-largest flow rate and of r as the flow

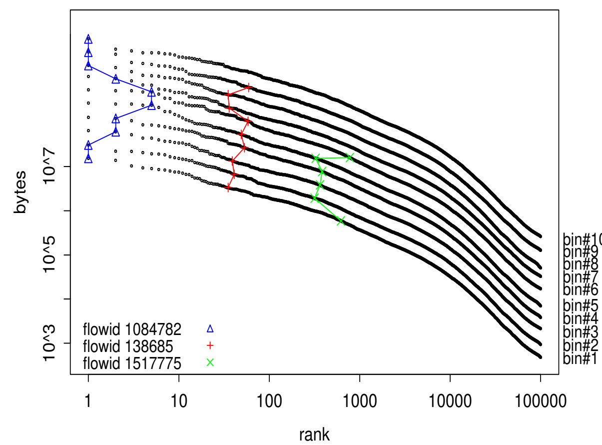


Figure 1: An illustration of Zipf's law across 10 successive time bins (raw IP flows, bin size = 1 minute).

rate's rank ($1 \leq r \leq n$), the relationship $rx_{(r)} = constant$ (or, more generally, $r^{\alpha} x_{(r)} = constant \ = c, \alpha > 0$) holds, at least approximately. When plotted on log-log scale, this *rank–size* relationship results in an (approximate) straight line with slope -1 or $-\alpha$, respectively, with the top-ranked flow rates being exceptionally large but rare and the lower-ranked rates being smaller but more common. This latter property follows directly from the *size–frequency* relationship that corresponds to Zipf's law and states that $f(x)$, the relative frequency of occurrence of a flow rate of size x satisfies the relation $f(x) = c \cdot x^{-2}$ (or, more generally, $f(x) = c \cdot x^{-(1+\alpha)}$), $x = 1, 2, \ldots$. To illustrate that Zipf's law applies bin-by-bin over time, Figure 1 shows the log-log plots of 10 size-rank relationships for flow rates of raw IP flows corresponding to 10 consecutive 1-minute bins. The plots are offset from one another by a small amount in the vertical direction to facilitate a visual assessment of Zipf's law across time, i.e., an approximate straight line behavior for each of the 10 plots. The same applies to other time periods, flow abstractions, and bin sizes (not shown). Flows whose flow rates appear in the top ranks (e.g., ranks 1–10) for at least one time interval or bin during their lifetime are referred to as "heavy hitters" or "elephants", while flows with consistently lowly-ranked flow rates (e.g., ranks beyond 100) are called "mice." We call flows exhibiting intermediate flow rates during their lifetime (e.g., ranks 11–100: never top-ranked, but not always lowly-ranked) "hybrids". In this paper we classify flows according to their ranks which are defined in terms of their absolute rate (e.g., greater than 1 Mbps) or

*This work is partially funded by DFG, Project 1126

[1] Something applies bin-by-bin if it is applicable for each individual bin.

their relative rate (e.g., greater than 10% of the traffic). Focusing on ranks facilitates the comparison of flows across time periods because the number of elephant flows and hybrid flows are constant. It can also be expected to add an element of stability, especially when the emphasis is on investigating the temporal dynamics of flow attributes such as "being an elephant flow".

Given that Zipf's law for flow rates applies on a per-bin basis across time begs the question whether a flow that lasts for a number of bins and has been classified as "heavy hitter" has earned this distinction because of being top-ranked only sporadically (i.e., the flow rates associated with just one or two bins made it into the top ranks) or persistently (i.e., the flow rates in most bins are top-ranked) throughout much of its lifetime. To illustrate, reconsider Figure 1 which shows log-log plots of 10 size-rank relationships for flow rates of raw IP flows corresponding to 10 consecutive 1-minute bins. Included in this figure are also three lines that connect various points on the different size-rank curves. These lines indicate how the ranks of the flow rates of three particular raw IP flows that were active during (parts of) this 10-minute interval changed over time. Note that while the left line shows about the same amount of movement in the rankings as the right one due to the log-scale on the x-axis, the number of ranks covered is far greater for hybrid and mice flows than for the elephant flows, with no indication that these flows will ever become elephants.

Understanding the persistency properties of how much traffic individual Internet flows (especially the heavy hitters) contribute during their lifetime to the overall traffic is important for traffic engineering. A commonly-used approach in traffic engineering targets the large flows primarily and attempts to optimize system performance mainly for them. Equally important is the ability to identify the causes underlying any observed persistency properties of these large Internet flows. Examples that follow this basic approach to traffic engineering include, among others, [6] (measurement support and accounting), [7] (Web server overload control), [8] (routing), [9] (scheduling), etc. Clearly, such approaches are more viable and effective if a substantial portion of the overall traffic in a bin is due to a few heavy hitters and if roughly the same cast of heavy hitters is responsible for a significant amount of the total traffic across different bins.

Unfortunately, studying the persistency aspects of Internet flows and demonstrating that the findings are representative requires a compromise concerning the available measurements. On the one hand, carefully examining persistency-related aspects of Internet flows is only possible using detailed packet-level traces, which tend to be collected in only a few places and for limited durations only. On the other hand, Netflow traces are more widely available and are therefore more suitable for checking whether or not certain findings are representative. However, because Netflow traces are in general too coarse-grained for investigating a number of dynamic aspects of individual flows, empirical studies relying on Netflow data often make the critical assumption that raw IP flows exhibit constant throughput (computed as flow size divided by flow duration) for the duration of their lifetime.

We thus use a combination of packet-level and Netflow measurements. We first rely on packet traces and Netflow traces derived from the same packet traces and check whether the assumption necessitated by the nature of the available Netflow data—that flows have constant throughput throughout their lifetime—is valid. If this assumption is invalid we may arrive at misleading or wrong conclusions about the persistency properties of Internet flows. We then use some large Netflow traces to illustrate the kind of persistency properties of measured Internet flows that are largely insensitive to the assumption of constant throughput.

Using our methodology for the available data, our initial findings depict a wide, yet largely unexplored spectrum of persistency-related behavior of Internet flow rates. For example, we find that heavy hitters at the level of raw IP flows show remarkable persistence and are likely to be top-ranked for the duration of their entire life. Thus, at the level of raw IP flows, the notion "once an elephant, always an elephant" holds with relatively high probability and suggests a simple heuristic for identifying heavy hitters—pick the top-N in each bin and "adjust" for those that last only one or so bins. Despite this persistence property of the heavy hitters, there seems to exist an unavoidable tradeoff between the need to consider a large number of top flows to account for a substantial portion of the overall traffic and the desire to account for only a few heavy hitters to ensure strong persistency properties. However, a formal statement concerning this tradeoff is beyond the scope of this paper. We also observed that while heavy hitters at the aggregate level are often made up of heavy hitters at the level of raw IP flows, we also encountered numerous instances where aggregate elephants contain essentially no raw IP flow elephants, but consist almost exclusively of raw IP flow mice.

The rest of the paper is organized as follows. After a brief discussion of related work in Section 2, Section 3 describes our proposed methodology for studying persistency-related aspects of Internet flows. In Sections 4 and 5 we describe the data sets that are used throughout the paper and validate our approach. Some of our initial findings are described in Section 6, where we focus in particular on the observed persistency properties of Internet flows under different time aggregation (i.e., different bin sizes) and different flow abstractions. We conclude in Section 7 by summarizing our experience and suggesting future research directions.

2. RELATED WORK

A common observation found in many measurement studies is that the sizes of raw IP or aggregated flows obey a Zipf-type law in the sense that a small percentage of the flows accounts for a large percentage of the total traffic (e.g., [8, 10, 11, 12] and references therein). As far as flow rates are concerned, a number of recent papers have attempted to characterize them and determine the causes of the observed rates at which flows transmit data in the Internet. In particular, [1] provides indirect evidence that a static version of Zipf's law for flow rates holds. In fact, Fig. 1 in [1] can be considered to express Zipf's law, with a bin size that corresponds to the length of the underlying trace data. In [2], the authors claim that a single high-rate flow typically accounts for much of the burstiness of the aggregate link traffic, and their Fig. 1(c) in support of this claim shows Zipf's law for flow rates (on linear-linear scale) for a single bin. While these and other papers make important contributions and improve our understanding of the nature of Internet flows and flow rates, none of them dwell on Zipf's law as such or on whether it holds on a per-bin basis across time.

To our knowledge, the first attempt at assessing the feasibility of identifying and isolating heavy hitters for traffic engineering purposes is reported in [13]. In this paper, the authors propose a definition of heavy hitters that accounts for both their volume and their persistence in time, and they rely on packet-level traces and corresponding BGP tables to examine the effectiveness of their proposed classification schemes for a fixed flow abstraction (determined by the BGP destination network prefixes) and different time aggregation (bin sizes of 1, 5, and 30 minutes). They find that while a single-feature classification scheme is impractical due to the large number of short-lived elephant flows it produces, a simple two-feature classification scheme that accounts for short transient dips or bursts is more successful in identifying the persistent heavy hitters. Our work adds to the original findings discussed in [13] by further exploring the persistence property of heavy hitters and by considering a fuller range of useful flow abstractions. At the same time, we move beyond the issues addressed in [13] by presenting an approach that is equally applicable to a collection of the top N flows (e.g., N=1000) as it is to the top-10 and that allows for a systematic investigation of potential causes underlying the collective behavior of groups of flows that deviates from "normal" or "typical" behavior (e.g., can "unusual" behavior be associated with standard or "emerging" applications?). However, perhaps the most important original contribution of our work is that it attests to the viability of traffic engineering approaches that trade off precise but scarce measurements (i.e., exact per-bin flow rates from fine-grained packet traces) for approximate but abundant information (i.e., constant throughput assumption for coarse-grained Netflow traces).

3. METHODOLOGY

When exploring the various persistency aspects of Internet flows, we divide time into bins and the basic information consists of the number of bytes or packets transmitted by the different flows in each bin. There are two methodological aspects to our work. First, we require detailed packet-level traces that can be easily and efficiently aggregated in time and across flows. Second, we rely on an effective multi-scale and multi-protocol analysis of the available data, where multi-scale refers to an ability to analyze the data at different time aggregations (i.e., bin sizes) and different flow abstractions, while multi-protocol implies the flexibility to slice the data by different protocols and applications.

In terms of flow abstractions, the packets belonging to a flow are usually determined by two parameters: *aggregation* and *timeout* [14, 15, 16, 17, 18]. Earlier work (e.g., [17]) has shown that the specific choice of the timeout parameter rarely changes the basic characteristics of the resulting flows. However, the aggregation parameter has a significant impact. To illustrate, *raw IP flows* are defined by the protocol they use as well as by source and destination IP addresses and port numbers. *Prefix flows* are defined by the source and destination prefix. The prefix for both the source and the destination IP address is computed from the IP address by using the mask from the longest prefix match in the routing table. If such a mask is not available we compute the prefix from the source and destination IP address using a fixed-length mask of length L. Note that a larger value of L implies a smaller degree of abstraction or aggregation.

To investigate temporal persistency aspects of Internet flows and their behavior across different time scales, we consider different bin sizes. To cover a reasonable range of interesting time scales, our choice of bin sizes ranges from 60 seconds to $120, 240, 480$ and 960 seconds. Since 60 seconds is significantly larger than the typical round-trip times experienced in the Internet [1], the impact of TCP-specific features (e.g., ACK spacing, window effects) on the resulting flow rates can be expected to be minimal and decrease further as the bin size increases.

Concerning the ability to restrict our analysis to specific applications and/or protocols, we note that UDP and TCP are likely to show different features. Similarly, different applications are known to have different signatures, e.g., distribution of flow length. Unfortunately the only way to associate flows with applications is via port numbers which is somewhat problematic.

3.1 Software Infrastructure

The methodology described above centers around computing per-flow throughput for different bin sizes, flow abstractions, protocols, and applications. To this end, based on the kind of input data, the first step consists of creating appropriate flows (see Section 3.2). Next, the resulting flows may have to be filtered by protocol and/or application before they can be used to form flows at different levels of aggregation. Depending on the input data and the desired abstraction level, one or more of the previous steps can be skipped. Relying on well-formed flows as input, the third step consists of computing the per-bin ranking of all the flows based on their per-bin throughput. Since we are mainly interested in the heavy hitters, we reduce the computational effort by considering only the top 5000 per-bin flow rankings and enter them into a database. This database contains all the information necessary for our study of the (non-)persistency of Internet flows across time, abstraction levels, protocols, and applications. For more details on the software infrastructure see Appendix A.

3.2 Input Data

Basically there are three kinds of network measurement data sources available. SNMP [19] data is the most available but unfortunately also the most coarse-grained data source and thus unsuited for our purposes. Another common data source are packet level traces. These provide sufficiently detailed information, but are not easily available from ISP backbones, or they usually cover only short timespans, e.g. a few hours. The third data source are flow-based measurements like *Cisco's Netflow* [20]. Flow-based measurements provide an acceptable level of detail for our purposes, providing byte counts and duration per flow and are more easily available than packet traces, especially for longer traces on ISP backbones. However, they pose a problem for our purposes, in that the time granularity of Netflow data is too coarse and requires approximations which in turn may lead to inaccuracies in the analysis and ultimately to incorrect conclusions. Cisco's NetFlow system is designed for traffic monitoring and accounting. Packets passing router interfaces on which NetFlow is enabled are aggregated in real time to unidirectional flows defined by source/destination IP and ports, protocol, and IP TOS bits. For each flow NetFlow gathers among others byte and packet counts, start and end timestamps as well as routing related information such as prefix mask lengths and AS numbers. Each router terminates flows according to a set of heuristics and then exports them to a NetFlow collector. A flow is terminated if it had no contributing packets for a certain amount of time (default 15 seconds). Flows that have been active for more than a certain amount of time (default 30 minutes) are terminated. This ensures that online-monitoring tools can operate on current information. Furthermore TCP flows are terminated whenever a FIN or RST packet is found. Finally, when the router runs short of memory, flows are terminated using undisclosed heuristics.

Since we are interested in per-bin packet or byte counts for each flow, the very nature of the available Netflow records makes it necessary to somehow distribute the total packet or byte aggregates per flow across the flow's lifetime. A natural choice is to assume

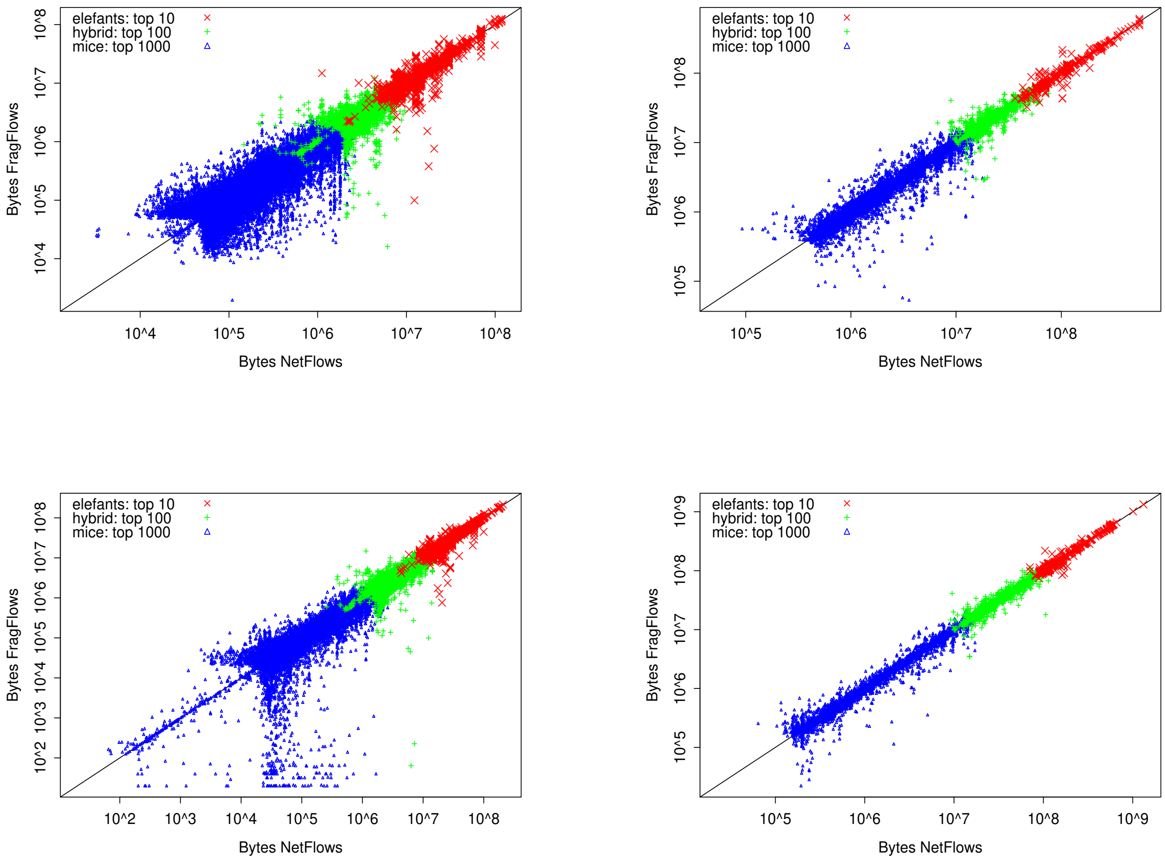


Figure 2: Scatterplots comparing the per-bin byte counts for NetFlows and FragFlows (top: raw IP flows, bottom: aggregated destination prefix flows using a fixed 16 **bit mask; left: bin size =** 60 **sec; right: bin size =** 480 **sec).**

a constant flow rate, computed as flow volume divided by flow duration. Practical experience with accounting and visualization systems suggests that this is a reasonable assumption, especially for reasonably large aggregation levels [21]. This is also consistent with the results of Barakat et al. [22] who model Internet traffic at the flow level via a Poisson shot noise process where the shape of the "shot" can be rectangular which corresponds to the assumption of constant rate. Additional complications with Netflow records are that NetFlow relies increasingly on sampling (see also Section 6.1) and that it uses several different timeout values. A router may expire a flow at any point if it needs to reclaim memory resources. The result of this process is that one flow may be split into several *raw flows*.

In this paper, we use both packet-level traces and raw Cisco Netflow traces. For the packet-level traces we reconstruct the individual flows and compute the corresponding exact per-bin rates called "fragments." We call the resulting flows *FragFlows*–they are defined in terms of the per-bin fragments which can vary across a flow's lifetime. For the Netflow traces we first recombine *raw flows* into flows and then compute their rates under the constant flow rate assumption. The resulting flows are called *NetFlows*–their per-bin flow rates are constant for the duration of a flow. Note that using packet-level traces, it is possible to reconstruct the appropriate Netflow counterparts that incorporate the constant flow rate assumption. These are also called *NetFlows*. In general the term *NetFlows* refers to flows whose per-bin rates are constant as a result of the constant flow rate assumption while *FragFlows* is synonymous with flows whose per-bin rates are computed exactly and are not likely to be constant.

Flows at different aggregation levels can be derived from both *FragFlows* and *NetFlows*. For an aggregate flow, its per-bin flow rates are simply the sum of the corresponding per-bin rates of those flows that make up the aggregate flow; for *FragFlows*, taking the sum involves the exact per-bin flow rates, while for *NetFlows*, the per-bin sum is taken over the average rates of the flows that are active during that bin and are part of the aggregate flow.

4. TRACES

We had access to several hour-long packet-level traces from the external Internet connection at the Universität des Saarlandes (UNI) and Leibnitz Rechenzentrum München (EDU). Both connections provide Internet access to a major university, some colleges, and several research institutes. The capacity of the UNI link was 155 Mbps, and the capacity of the EDU link was 622 Mbps. In both locations the recording was done via the monitoring port of a Gigabit Ethernet switch just before the traffic passes the last router to the Internet. In terms of Cisco Netflow traces we had access to several days worth of traces collected at different backbone routers of a Tier-1 ISP.

Throughout this paper we use the following data sets. The packet trace P1 was gathered at the EDU location using a packet filter that captured only traffic to and from the CS department. The trace consists of a total of 349,194,384 packets (more than 10.3 GB of compressed data) and was collected on Friday, Oct. 31, 2003, 11:17-

15:22. A second data set P2 was gathered at the same location, but without the packet filter used for P1 (i.e. this trace contains all packets that crossed the monitored link). Trace collection started on Wednesday, Nov. 13, 2002, 19:10 and ended on Thursday morning at 02:43. The trace consists of 344,965,957 packets or more than 11 GB of compressed data. The third trace P3 consists of 104,583,280 packets (some 2.5 GB of compressed data) and was collected at the UNI location on Tuesday, Feb. 02, 2003, between 12:00–15:29.

The Cisco Netflow trace F1 was collected from a single backbone router within a Tier-1 ISP. The trace contains a day worth of Netflow data, collected on Dec. 11, 2001. The data set F1 contains over 211 million flow records or about 4 GB of compressed data. This Netflow trace is unsampled and has a loss rate of 9% (mostly due to the capacity limits in the monitoring infrastructure, not in the ISPs infrastructure). A second day-long sampled Netflow trace F2 was collected on Sept. 5, 2002 and consists of almost 330 million flow records or more than 5.1 GBytes of compressed data.

5. VALIDATION

Before embarking on a Netflow-based study of the persistency aspects of Internet flows, we first examine in this section the validity of the constant flow rate assumption. As discussed earlier, this assumption is unavoidable when using Netflow data in this context. To this end, we rely on our packet-level traces for which we can derive both FragFlows as well as NetFlows. After obtaining the per-bin rankings for the top 5000 flows for both NetFlows and FragFlows at various abstraction levels, as well as for different protocols and applications, we match the appropriate FragFlows and NetFlows and compare them to assess the impact of the constant throughput assumption on the validity and quality of our findings. More precisely, we select the top 1000 entries from each bin and located the matching counterpart if it exists among the top 5000 entries.

5.1 Per-bin byte differences

We start by comparing how the per-bin byte counts of the FragFlows differ from those of the NetFlows for different time aggregations and flow abstractions. The working hypothesis is that we should expect differences, but that they will diminish as we consider larger bin sizes and/or flow aggregates. In this context, one of the objectives is to try and identify the causes of and quantify to some extent the expected differences for small bin sizes and raw IP flows.

The impact of the constant flow rate assumption for NetFlows can be expected to be most dramatic when comparing how many bytes a FragFlow is contributing to a particular bin and how many bytes the corresponding NetFlow contributes. To illustrate this comparison, Fig. 2 shows scatterplots of the per-bin contributions of NetFlows (x-axis, log-scale) against the per-bin contributions of the corresponding FragFlows (y-axis, log-scale) for the trace P1. The top row is for raw IP flows and bin sizes of 60 seconds (left) and 480 seconds (right), while the bottom row is for aggregated destination prefix flows using a fixed 16 bit mask and the same two bin sizes. Note that the plot only shows distinct points; duplicates are removed before plotting. We use different symbols to indicate the ranking that a particular point is associated with. A small "△" corresponds to a byte count that has one ranking (FragFlow or NetFlow) in the top 1000 but none in the top 100. A "+" marks those byte counts that have one ranking in the top 100 but not top 10,

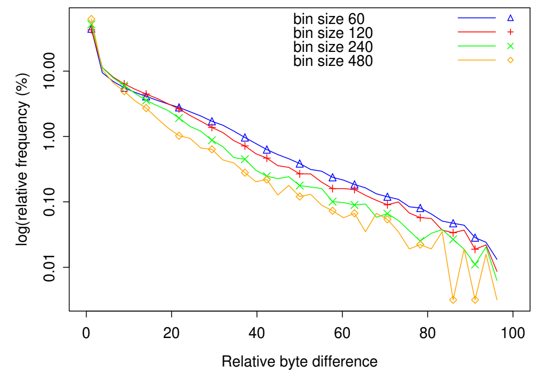


Figure 3: Histogram plot of the relative byte differences between FragFlows and NetFlows with linear x- and logarithmic y-axis (raw IP flows, bin size = 60**,** 120**,** 240**,** 480 **sec).**

and a "×" identifies the byte counts that have a top 10 ranking. The most pronounced feature in all of these plots is a strong concentration of the points around the diagonal, with varying degrees of deviation as we consider different bin sizes and/or flow abstractions.

In relative terms, these deviations from the diagonal seem to be smallest for byte counts with a top 10 ranking and tend to get larger as we consider more of the more frequently occurring lower-ranked byte counts. Also, as we consider either larger bin sizes (left to right in Fig. 2) or larger aggregation levels (top to bottom), or a combination of larger bins and aggregates (top left to bottom right), the concentration along the diagonal is accentuated. As far as increasing the bin size is concerned, one explanation for this observation is that more flows will completely fall within a single bin, and that this feature impacts not only the many lowly-ranked flows whose durations tend to be shorter than those of the few top-ranked flows. In terms of increasing the level of flow aggregation, the variability of the per-bin byte counts of large aggregates is bound to decrease as a predicted by the Central Limit Theorem.

To quantify the degree of (in)accuracy of the approximation resulting from the constant flow rate assumption, we compute for each per-bin byte count the relative byte difference between the FragFlow and the NetFlow entries. This is done for each bin and flow by taking the absolute value of the difference between the two byte counts, dividing it by the maximum of those two values, and multiplying by 100 to get percentages. Fig. 3 shows histogram plots of the relative byte differences for different bin sizes and illustrates that the quality of the constant throughput assumption increases with bin size. Similar conclusions can be drawn when considering the same histogram plots (not shown here) for different flow aggregation levels. Overall we observe that—as expected—the accuracy of using the more widely available NetFlows instead of the hard-to-come-by FragFlows increases with bin size and with aggregation level, implying that the constant throughput assumption may be appropriate for certain flow abstractions.

While concentration around the diagonal in the plots in Fig. 2 is highly desirable, points that clearly deviate may also be informative, especially if they concern top-ranked byte counts, and deserve

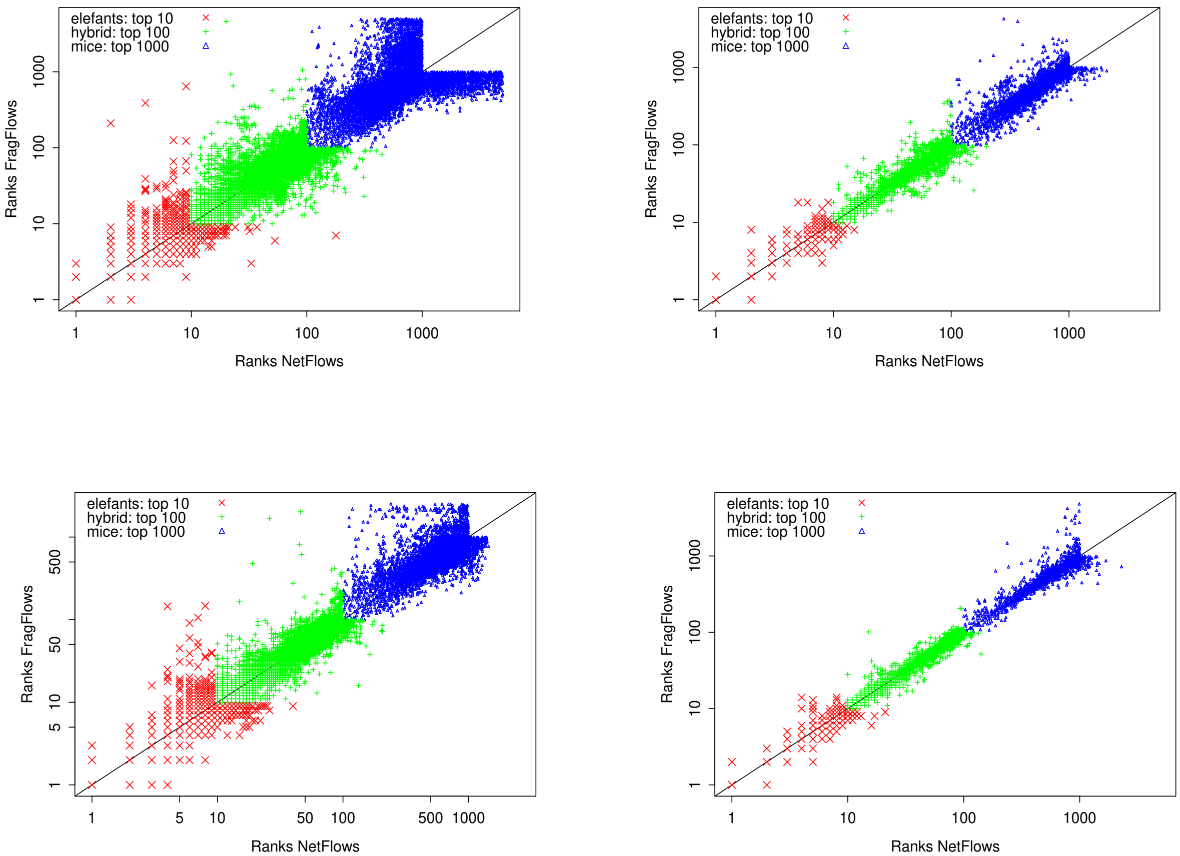


Figure 4: Scatterplots comparing the per-bin byte count ranks for NetFlows and FragFlows (top: raw IP flows, bottom: destination prefix flows using a fixed 16 bit mask; left: bin size = 60 sec; right: bin size = 480 sec).

closer inspection. [2] For example, in the top left plot in Fig. 2, we can identify 18 bins associated with 16 flows where the difference in FragFlow- vs. NetFlow-derived byte counts was large enough to cause the byte counts to be classified as top 10 for FragFlow and as top 100 for NetFlow, or vice versa. Of these 18 "outliers", 14 occurred at the start (6) or the end (8) of the flows. Possible explanations include: TCP slowstart and initial protocol overhead for those at the beginning, and timeout effects for those at the end.

5.2 Per-bin rank differences

Next we examine what impact the observed differences in per-bin byte counts that are the result of the constant throughput assumption have on the per-bin ranking of the flows.

While the previous subsection focused on the impact of the constant flow rate assumption on byte counts, we now examine the differences that are imposed by this assumption on the ranking. The raw rank data derived from the P1 trace are given in Fig. 4 which shows scatterplots of the per-bin NetFlows-derived ranks (x-axis, log-scale) against the corresponding FragFlows-derived ranks (y-axis, log-scale). As in Fig. 2, the top row is for raw IP flows and bin sizes of 60 seconds (left) and 480 seconds (right), while the bottom row is for aggregated flows using a fixed 16 bit mask and the same two bin sizes. The symbols have the same meaning as in Fig. 2, but note that now, the top-ranked per-bin flow rates are concentrated in the lower left rather than in the upper right corners of the four plots. After accounting for the artifacts caused by selecting only the top 1000 entries and by using log-scale in conjunction with ranks that can only take discrete values, the common dominant feature in these plots is again a pronounced concentration of the points around the diagonal, with some obvious deviations. However, these deviations from the diagonal diminish significantly as either larger bin sizes or larger flow aggregates are considered.

Before addressing the issue how the constant flow rate assumption impacts the ranking of individual flows, we first examine how much of a per-bin rank difference can be expected as a result of a given per-bin byte count difference. In effect, in answering this question, we combine the information from Figs. 2 and 4 to generate Fig. 5. More precisely, to generate the relevant information, we consider the relative per-bin byte count differences instead of their absolute values because generally, for elephants, larger absolute byte changes are needed to switch ranks than for mice. At the same time, in terms of rank differences, it seems more sensible to consider absolute rather than relative rank changes. To be able to examine the relative byte differences in conjunction with the smaller absolute rank differences in more detail, we manipulate the data by adding a constant offset of 0.1 to both the absolute rank differences as well as to the relative byte differences; we also introduce some jitter to the absolute rank differences by adding a uniform random amount between 0 and 0.4 to each absolute rank difference so as to avoid the situation that all points with the same

[2] An obvious artifact in Fig. 2 are the vertical bands that are associated with one and the same NetFlow byte count and result from having in general many different FragFlow byte counts for the same flow.

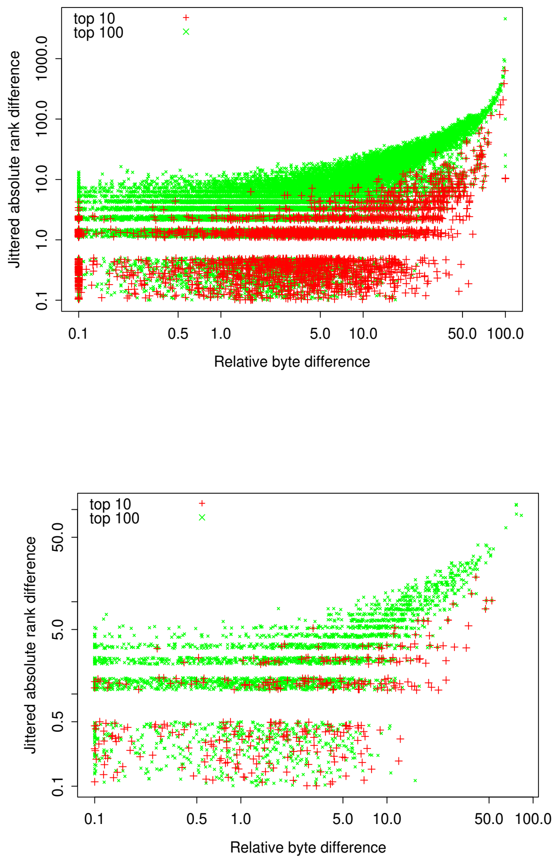


Figure 5: Scatterplots of the relative per-bin byte count differences between FragFlows and NetFlows vs. jittered absolute per-bin rank differences between FragFlows and NetFlows (top: raw IP flows, bin size = 60 sec; bottom: aggregated destination prefix flows using a fixed 16 bit mask, bin size = 480 sec).

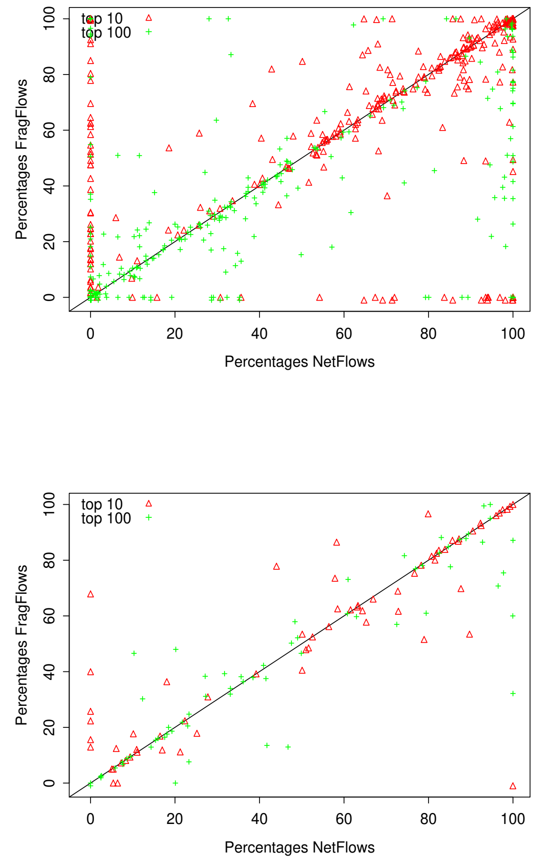


Figure 6: Scatterplots of percentages of bytes from elephant/hybrid bins for NetFlow-derived elephants/hybrids vs. percentages of bytes from elephant/hybrid bins for FragFlows-derived elephants/hybrids (top: raw IP flows, bin size = 60 sec; bottom: aggregated destination prefix flows using a fixed 16 bit mask, bin size = 480 sec).

(integer-valued) rank difference appear as a single point in the plot. The resulting two plots (one for raw IP flows and a bin size of 60 sec, and one for aggregated flows using a fixed 16 bit mask and a bin size of 480 sec) are shown in Fig. 5 and correspond to the top left and bottom right plots shown in Figs. 2 and 4.

The clearly visible band with (jittered) rank differences between 0.1 and 0.5 corresponds to those bins where the FragFlows and the corresponding NetFlows entries have the same rank. It is interesting to note that in the non-aggregated case (top plot), rather large relative byte difference (up to 50%) can occur without influencing the rank too much. Once we include the next few discernible bands corresponding to rank differences of ± 1, ± 2, to ± 5 or so, the number of top-ranked bins is drastically reduced, more so for the aggregated case (bottom plot) than for the non-aggregated example. The remaining elephant bins are the ones where a large relative byte difference leads to a relatively large rank change. Fig. 5 begs the question how a flow rate can more or less keep its rank in going from NetFlows to FragFlows even if the byte count difference is relatively large. There are (at least) two arguments that can be put forward. For one, the byte difference may not be big enough to either reach the byte count of the next higher ranked entry or let it drop below the next lower ranked entry. Alternatively, another flow that was lower or higher ranked than the current one has a large relative byte difference and is therefore now ranked higher or lower than the current one. The latter argument also explains why a flow may change its rank in a bin even if the byte difference is zero or extremely small. Similar comments apply when considering different bin sizes and/or flow aggregation levels. Fig. 5 also illustrates that because the byte differences between the individual ranks are much smaller for the lower-ranked entries, the observed rank differences for the latter will be larger than for the top-ranked items.

5.3 Per-flow rank differences

Until now, we have been mainly concerned with the per-bin byte count differences between FragFlows and NetFlows and with their impact on the resulting per-bin rank differences. Here we will use the insight gained so far at the per-bin level and apply it to determine the impact that the constant throughput assumption has on the flows as a whole. To this end, we focus on the heavy hitters, where we define a heavy hitter to be a flow that is ever ranked an "elephant" (i.e., in the top-10) in any bin during its lifetime. Other choices of defining an elephant (e.g., in the top-5, or top-20) yield similar results. For such flows we are interested in determining what percentage of the total bytes contributed by an "elephant" flow can be attributed to bins that are ranked within the top-10 or the top-100. Accordingly, a flow is considered a "hybrid" flow if its top ranked bin is a "hybrid" (i.e., in the top-100, but not in the top-10). Fig. 6 shows two scatterplots of the percentage of bytes contributed by an elephant or hybrid during bins that were ranked within the top-10 (for elephants) or top-100 but not top-10 (for hybrids) using NetFlows (x-axis) against the same FragFlow-derived quantity (y-axis). The top plot deals with the non-aggregated case (i.e., raw IP flows and a bin size of 60), and the bottom plot is for the aggregated case (i.e., aggregated flows using a fixed 16 bit mask and a bin size of 480 sec).

Fig. 6 shows a number of informative properties related to relying on NetFlows as compared to FragFlows. For one, most of the points in both plots scatter around the diagonal, some are right on the diagonal (indicating a perfect match between NetFlows and FragFlows), some occupy the line $x = 0$ (vertical line through 0) and others the line $y = 0$ (horizontal line through 0). Looking first into the 24 (total points in the top plot is 524) flows satisfying $y = 0$, we find that the median distance of the FragFlow and the NetFlow ranking for the bins that cause each of these flows to be considered an elephant is 1 (mean is 1.8). This suggests that NetFlow just barely overestimated the ranking in comparison with FragFlow. Unfortunately, these edge effects cannot be avoided whenever one chooses a simple static elephant classification such as top-10 ranking, but time aggregation helps in alleviating this problem. Next the 51 flows satisfying $x = 0$ have little to do with edge effects, but represent in some sense the price one has to pay when using NetFlows instead of FragFlows in classifying Internet flows. Indeed, the reason for this drastic mismatch in this case between NetFlows and FragFlows is that while FragFlow is capable of capturing the dynamics of the within-flow data exchange, these details are invisible to NetFlows. To illustrate, a sample sequence of per-bin ranks for a FragFlow is: 135, 9, 11, 7, 15, 18, 18, 19, 17; the Netflow-derived per-bin rank sequence for that same flow is: 92, 13, 12, 15, 15, 18, 18, 19, 17. One consequence of NetFlows "mis-classifying" some elephant bins as hybrid bins is that the affected flows tend to get a large percentage of their bytes from hybrid bins when the actual (FragFlow-derived) percentage is in fact smaller. This explains the set of hybrid flows clustering around the line $x = 100\%$. In general, this problem can be alleviated with flow aggregation, which illustrates yet again that aggregation is the proper tool for achieving a desirable degree of accuracy when using NetFlows instead of FragFlows .

6. MULTI-SCALE/PROTOCOL FLOW ANALYSIS

The findings reported in the previous section justify the use of the widely available but coarse-grained unsampled Netflow data in conjunction with the constant flow rate assumption for studying persistency related aspects of Internet flows, especially of the heavy hitters. While the constant bandwidth assumption inherently gives raise to inaccuracies and errors, they can often be controlled by using aggregation and focusing on the large flows. Using the available large Netflow data sets, we illustrate in this section the kind of multi-scale and multi-protocol flow analysis that is intended to shed light on various persistency related aspects of Internet flows. In Section 6.1 we start out using both unsampled as well as sampled NetFlow traces and illustrate that there are qualitative differences in the results. Because of this observation and since the previous sections only justify the use of unsampled NetFlow traces, in the remaining part of this paper we focus exclusively on unsampled traces and leave the detailed exploration of the impact of sampling for future work. Our findings are largely qualitative, but they are also representative to the degree that we confirmed them using other traces. These latter data sets have also been used to validate on a case-by-case basis some of the findings presented below.

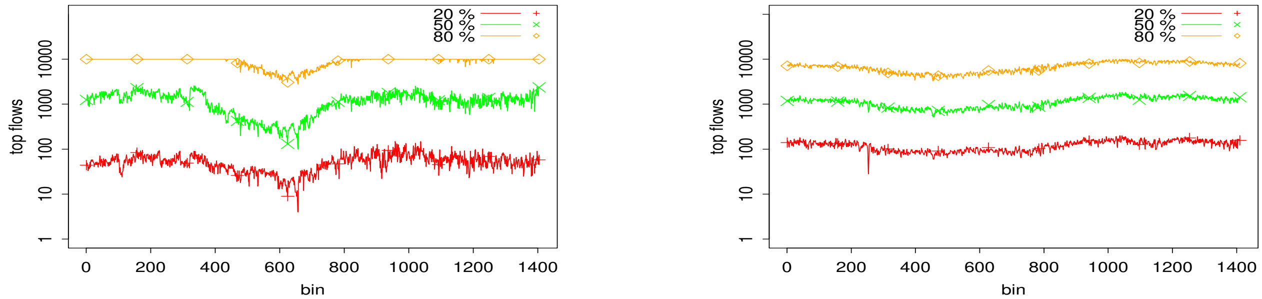
6.1 On the dynamics of flow rankings

Informally, Zipf's law and its variations are often interpreted as 80-20 or 90-10 rules, which state that some 80% (or 90%) of consequences stem from some 20% (or 10%) of causes. In the present context, this translates into "a significant portion of the total number of bytes in a bin is due to a relatively small percentage of the top-ranked flows (i.e., flows with the highest flow rates). Relying on the unsampled Netflow trace, F1, and the sampled Netflow trace, F2, Fig. 7 considers raw IP flows and 1-minute bins and shows how many of the top-ranked flows are needed for each bin to account for 20%, 50%, and 80% of the bin's total traffic volume. For example, we note that the top 100 or so flows account for some 20% of the total traffic per bin, the top 1,000 or so flows are responsible for about 50%, and to account for 80% of the total traffic, we need to consider way more than the top 1,000 flows (e.g., there are times that require more than the top 10,000 flows). We note that the relative bytes per top ranked flows for the unsampled Netflow trace, F1, is larger than those for the sampled Netflow trace, F2. Yet for the lower ranked flows they are less. Also note that the overall per bin volume of the unsampled trace is about a factor 2-3 smaller than for the sampled trace.

Fig. 7 leaves open the possibility that the cast of top-ranked flows can vary considerably from one bin to the next; that is, due to the arrivals of new and the departure of existing flows, there always exist opportunities for newly arriving flows to make it into the top ranks and for existing flows to fall out of the top ranks. Note that for raw flows, whose rates are by definition constant throughout their lifetimes, the arrival/departure dynamics of flows is the only ingredient that can cause instability among the top-ranked flows (i.e., significant rank changes from one bin to the next). In contrast, aggregate flows can also change ranks as a result of fluctuations in their rates from one bin to the next due to the arrival/departure dynamics of their constituent flows.

Fig. 7 also leaves open the possibility that the dynamics is only due to churn; that is, there is no persistency of flows across bins and the observations automatically follow from the well-known heavy-tailed distribution of flow sizes or lengths. However, a simple comparison of the flow length distributions shows that this is not the case for the top ranked flows. For example, for the trace F1 the median of the flow length distribution for 60 second bins increases from 4.2 seconds for the top 10000 flows to 44.7 for the hybrid flows and to 57.8 seconds for the elephant flows. In the case of 480 second bins, the larger bin length improves the ability of longer but lower average rate flows to be higher ranked. In fact, for the same trace, the median of the flow length distribution changes from 44.7 seconds to 136.0 seconds for hybrid and from 57.8 to 296.5 seconds for elephants. The maximum flow length coincides with the trace duration.

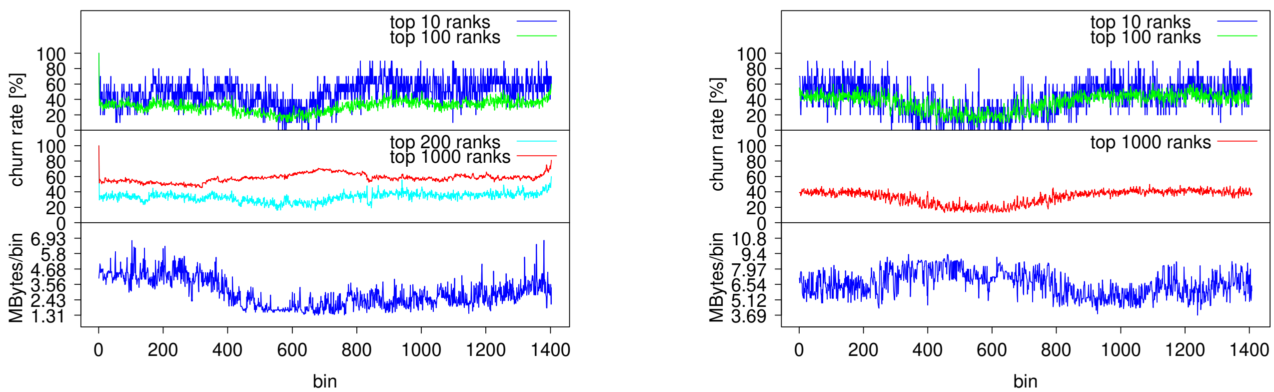
To illustrate the degree of (non-)persistency among the top-ranked flows over time, Fig. 8 considers raw IP flows and 1-minute bins again and shows the "churn rate" among the top 10, top 100, and top 1000 flows, respectively. Here, for each bin, the churn rate is defined to be the percentage of top 10 (top 100, top 1000) flows



a) non-sampled

b) sampled

Figure 7: Number of top-ranked flows needed to account for a given portion of the total traffic per bin (raw IP flows, bin size = 60 **sec).**



a) non-sampled

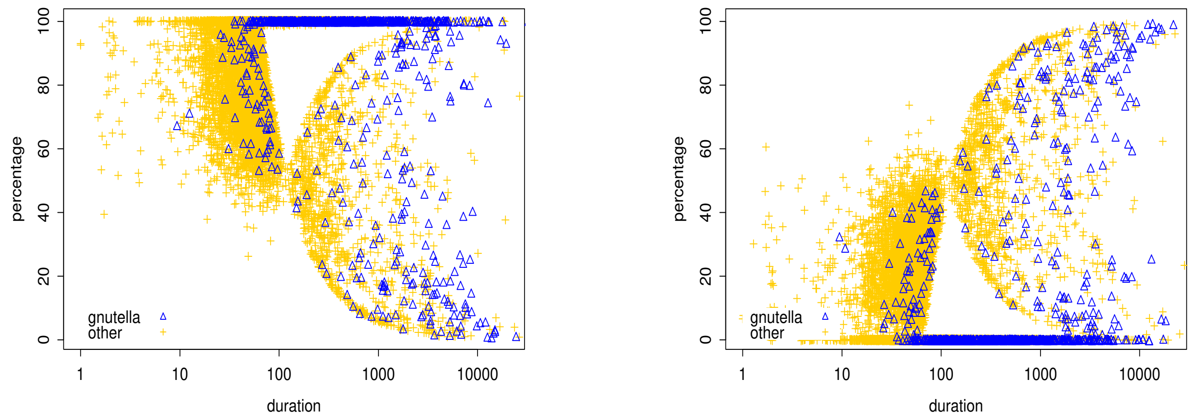
b) sampled

Figure 8: Upper and middle parts: Churn rate processes associated with the top 10, top 100, top 200, and top 1000 flows, respectively. Lower part: Time series of flow rates needed for a newly arriving flow to move into the top 10. (Raw IP flows, bin size = 60 **sec.)**

in that bin that were not among the top 10 (top 100, top 1000) flows in any of the previous bins. Thus, a high churn rate is an indication of significant non-persistency among the top-ranked flows, while a low churn rate reflects a considerable degree of persistency among the cast of top-ranked flows in time. In the upper and middle parts of Figures 8 a) and 8 b) we can see that while the different churn rates are roughly comparable, they show subtle but nevertheless important differences. Overall, the variability of the churn rate drops as one considers more ranks. For F1 (Figure 8 a)), the churn rate among the top 100 and top 200 ranks is lower than the churn rate for the top 10 ranks. During the early morning hours (bins 500-900 or so) this difference is reduced. The churn rate for the top 1000 ranks shows the opposite behavior. It is larger than the churn rate for the other ranks and it increases as the traffic volume decreases (not shown). This indicates that most flows that are in the top 1000 but not the top 200 are short lived and interchangeable. The byte volume differences between flows at rank 1000 are in the order of 10s of bytes. For F2, the churn rate among the top 100 and top 1000 ranks is slightly lower than the churn rate for the top 10 ranks during the early AM hours (bins 1–300 or so), it is slightly higher during the later AM hours/early PM hours (bins 300–900 or so), and appears to revert to the early AM behavior during the late PM hours (bins 900–1440). For this trace the churn rate for top 1000 ranks does not show the opposite behavior as for F1 since the byte volume differences between flows at rank 1000 are in the order of 100s of bytes.

For both, F1 and F2, the churn rate appears to be correlated with the total number of flows, see Figure 7. However during the less busy periods in F1, a flow needs to contribute a smaller number of bytes to a bin in order to be top ranked than during the corresponding periods in F2. For a visual assessment of this observed behavior of the churn rates, we show in the lower parts of Fig. 8 the time series representing for each bin the rate (i.e., number of bytes per bin) at which a newly arriving flow would have to send data to move into the top 10 ranks (i.e., be classified as heavy hitter). The differences observed in the plots in Fig. 8 resulting from the use of unsampled vs. sampled Netflow traces clearly warrants further investigations. For the rest of this paper we focus on the unsampled Netflow trace F1.

Given the persistency behavior of Internet flows suggested by Fig. 8, we next focus on the heavy hitters, where we define a heavy hitter as in Section 5 and ask whether or not heavy hitters have a distinct persistency property. Put differently, we are interested in whether "once an elephant" implies "always an elephant," at least with high probability. Evidence of such persistency properties for the largest Internet flows is crucial for approaches to traffic engineering that rely on the persistence in time of flows to remain elephants. For the trace F1 with a bin size of 1 minute and when



a) Heavy hitters as Elephants

b) Heavy hitters as Hybrids

Figure 9: Scatterplots of lifetimes of heavy hitters (log-scale on x-axis) against relative amount of time spent as elephants (left plot) or hybrids (right plot) for NetFlow trace F1**.**

considering raw IP flows, we extracted a total of 5,666 heavy hitters and show in Fig. 9 scatterplots of the heavy hitters' lifetimes against the percentage of time they were ranked elephants (ranks 1-10; left plot) or "hybrids" (ranks 11-100; right plot); we use log-scale for their lifetimes on the x-axis and linear scale for percentages on the y-axis. To avoid certain artifacts due to binning when computing the percentage of time flows were ranked elephants/hybrids, for flows that start during some bin and subsequently cover one or more full bins, the time spent in the beginning partial bin is counted towards the flow's ranking in the first full bin; ending partial bins are handled similarly.

Fig. 9 reveals a number of interesting features as far as the heavy hitters are concerned. Ignoring for the time being the coding of the points, we first note that about $1/2$ of the heavy hitters are elephants during their entire lifetime (i.e., out of a total of 5,666 points, some 2,875 fall on the $y = 100\%$ line). Second, heavy hitters who are alive for 2 or more bins and are not elephants during their whole lifetime have about a 40% chance to be elephants for more than half their lifetime and a 60% chance to be elephants for less than half their lifetime (i.e., half-moon shaped cluster starting at $x = 120$ sec and $y = 50\%$). Finally, when comparing the left and right part of figure 9, the anti-symmetry between heavy hitters as elephants and heavy hitters as hybrids is not an accident. In fact, the right plot shows some 60% of the heavy hitters are never hybrids, and those heavy hitters that are alive for 2 or more bins and are hybrids for some time have about a 60% chance to be hybrids for less than half of their lifetime. Note that the remaining structure in the left upper (left lower) corner of the left (right) plot of Fig. 9 is relatively uninteresting since those points correspond to heavy hitters that are alive for less than 120 seconds (2 bins) [3]. In summary, Fig. 9 shows that more than 95% of the heavy hitters are elephants for more than half of their lifetime. A breakdown of all the heavy hitters by application is easily possible, but simply shows the usual suspects (e.g., web, nntp, p2p, ftp, and others) and individually, they produce plots similar to the ones shown in Fig. 9. To illustrate, we use in Fig. 9 the symbol "$\triangle$" to denote heavy hitters associated with the well known p2p application Gnutella.

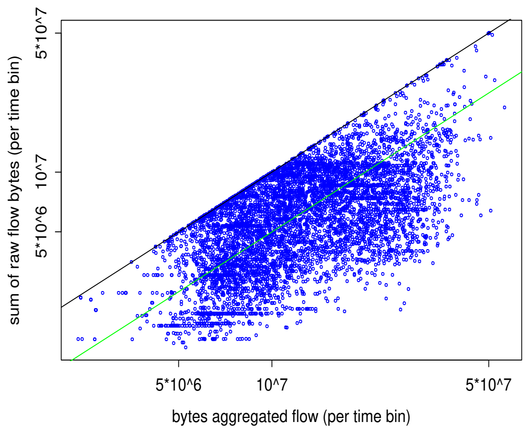
[3]Flows that last less than 120 seconds and are elephant for only part of their lifetime are split across two bins. Since they are more likely to be elephants in the bin in which they spend most of their time it is not surprising that most of the percentages are larger than 50%.

6.2 Heavy hitters and time aggregation

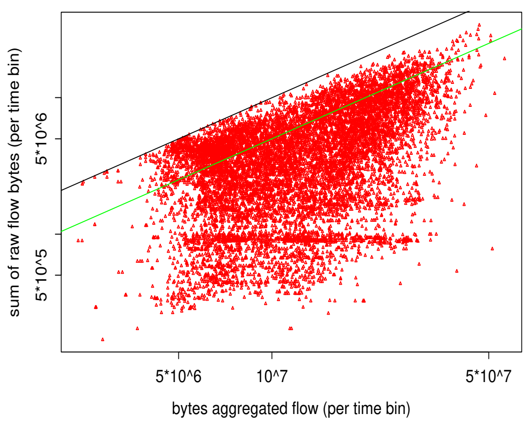
Part of the multi-scale aspect of our flow analysis involves considering different time scales and performing the same type of persistency study across a range of time scales. Our analysis (not shown here) suggests that the observations reported in Section 6.1 are largely invariant under different choices of bin sizes and hold in a genuinely multi-scale fashion. To explain this property, note that heavy hitters at the level of raw IP flows and at large time scale tend to remain heavy hitters at finer time scales. In fact, considering for example coarse scale to mean an 8-minute bin size and fine scale to mean a 4-minute bin size, a raw IP flow that is a heavy hitter at coarse scale will simply re-distribute the bytes in each large bin evenly among the two corresponding 4-minute bins at the finer time scale and is thus likely to cause the resulting flow to be a heavy hitter at the finer time scale. The situation is slightly more complicated for heavy hitters at the aggregate level (e.g., prefix flows), because their contributions to a large bin are generally no longer distributed evenly among the corresponding smaller bins at the finer time scale. Nevertheless, a preliminary analysis of the aggregate heavy hitters across a limited range of time scales (from a few seconds to hundreds of seconds) shows that heavy hitters tend to be invariant under time (dis)aggregation which explains why certain persistency properties associated with heavy hitters can already be gleaned from an analysis at coarse time scales which generally involves a substantially reduced data set and is therefore faster and more efficient.

6.3 Heavy hitters and flow aggregation

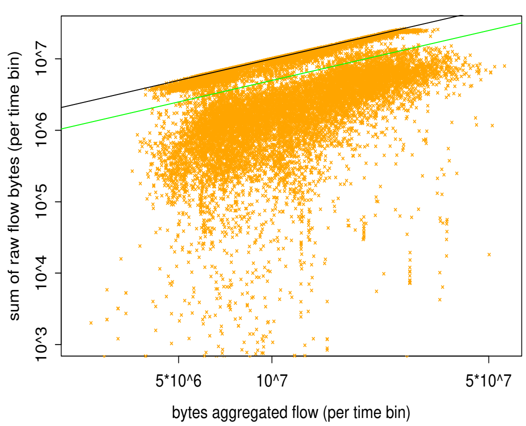
Another aspect of our multi-scale analysis of Internet flows concerns aggregation in IP or flow abstraction. That is, Netflow data lends itself naturally to different levels of aggregation, from raw IP flows (defined by source and destination IP addresses and port numbers and protocol) to prefix flows (defined by source and destination prefix) to AS flows (defined by source and destination AS). To illustrate that flow aggregation is in many ways more intricate than time aggregation, we explore for trace F1 in Fig. 10 the question whether or not flows that are heavy hitters at some coarse scale of flow aggregation (e.g., prefix flows) are in general made up of constituents that are heavy hitters at a finer scale of flow abstrac-



a) Elephant Contributers, non-sampled



b) Hybrid Contributers, non-sampled



c) Mice Contributers, non-sampled

Figure 10: Scatterplots of flow rates of aggregate heavy hitters (log-scale on x-axis) against aggregate contributions of the constituent raw IP flow elephants (top), hybrids (middle), and mice (bottom) for non-sampled Netflow trace F1 **(Bin size = 1 minute).**

tion (e.g., raw IP flows). That is, what is the observed behavior of heavy hitters under flow (dis)aggregation? To this end, for a 1-minute bin size, Fig. 10 shows scatterplots of the per-bin contributions of the heavy hitters at the aggregate level (destination-prefix, log-scale on x-axis) against the sum of the per-bin contributions of those flows that were elephants (top row), hybrids (middle row), and mice (bottom row), respectively, at the level of raw IP flows (log-scale on y-axis). We observe that while there are a number of points between the two lines $y = x$ and $y = x/2$ in the top and middle rows, the vast majority of points in the bottom row are concentrated in that area. That is, while many of the aggregate heavy hitters are largely (e.g., more than 50%) made up of raw IP flows that are elephants and hybrids, respectively, the bottom row shows a large number of instances where the aggregate heavy hitters are almost exclusively the result of raw IP flows that are mice. It would be interesting to explore this finding further and check for example whether or not the latter aggregates correspond to particular Web servers and whether or not the former can be associated with particular applications, but we leave such questions for future work.

7. SUMMARY

Using a combination of packet-level and Netflow traces, we examined the quality and accuracy of results obtained from relying on Netflow data (including the widely-assumed constant flow rate assumption) instead of packet-level data for studying various aspects of Internet flows. Subsequently, we focus on a largely unexplored facet of Internet traffic analysis related to Zipf's law for IP flow rates as a function of time and to the persistency properties of Internet flows, especially of the large flows or "elephants." Several applications motivate us ranging from reservation to traffic engineering. Our examination allows us to make a number of interesting observations. First, for all practical purposes, using the widely available but relatively coarse-grained Netflow traces vs. the scarcely recorded but very detailed packet-level traces for studying properties of Internet flows is justified. Errors and mismatches due to the constant bandwidth assumption underlying the use of Netflow traces are always a concern, can in general be significantly reduced by aggregation (time and/or flow aggregation) or by focusing on the heavy hitters, and should be dealt with on a case-to-case basis. Second, at the level of raw IP flows, elephants tend to stay as elephants for a very large proportion of their lifetime and mice rarely move beyond their category; at the level of aggregate flows, the persistency properties tend to be more intricate due to a richer set of possible causes for variations under time or flow aggregation. Our software allows for examination of such phenomena in isolation (raw flows), as well as a variety of aggregations (prefixes) and de-aggregations (traffic partitioned into component protocols), and varying time scales.

The work presented in this paper identifies a number of issues that deserve further attention. For the purpose of reducing the volume of Netflow measurements, the more recently collected Netflow traces represent sampled data. As observed, sampling is yet another source of potential inaccuracy that needs to be dealt with when checking the quality and accuracy of findings that rely on sampled Netflow traces. Clearly, the validation approach presented in this paper can be extended to handle sampled Netflows. Another direction for future work is motivated by a number of the plots presented in this paper, each of which identifies its own set of "interesting" flows (i.e., clearly identifiable "outliers") that beg for a full-blown use of our proposed methodology, including a "drilling down" into the protocol and/or application-specific aspects that can be gleaned from the data. Such a detailed multi-scale and multi-protocol analysis (including a systematic study of prefix-based flows) is part of future work. Furthermore "joining" this kind of data with other relevant measurements such as appropriate BGP routing tables promises additional insights.

Acknowledgments

We would like to thank AT&T Labs Research for providing access to Netflow data and Leibnitz Rechenzentrum, München for their help in collecting packet traces.

8. REFERENCES

[1] Y. Zhang, L. Breslau, V. Paxson, and S. Shenker, "On the characteristics and origins of Internet flow rates," in *Proc. ACM SIGCOMM*, 2002.

[2] S. Sarvotham, R. Riedi, and R. Baraniuk, "Connection-level analysis and modeling of network traffic," in *Proc. ACM Internet Measurement Workshop*, 2001.

[3] S. Uhlig and O. Bonaventure, "Implications of interdomain traffic characteristics on traffic engineering," Tech. Rep. Infonet-TR-2001-08, University of Namur, Belgium, 2001.

[4] G. Zipf, *Human Behavior and the Principle of Least Effort*. Addison-Wesley, 1949.

[5] M. Mitzenmacher, "A brief history of generative models for power law and lognormal distributions.," *Internet Mathematics*, 2003.

[6] C. Estan and G. Varghese, "New directions in traffic measurement and accounting," in *Proc. ACM SIGCOMM*, 2002.

[7] J. Jung, B. Krishnamurthy, and M. Rabinovich, "Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDNs and Web Sites," in *Proc. of the World Wide Web Conference*, 2002.

[8] W. Fang and L. Peterson, "Inter-AS traffic patterns and their implications," in *Proc. of the 4th Global Internet Symposium*, 1999.

[9] L. Guo and I. Matta, "The war between mice and elephants," in *Proceedings of ICNP*, 2001.

[10] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True, "Deriving Traffic Demands for Operational IP Networks: Methodology and Experience," in *Proc. ACM SIGCOMM*, 2000.

[11] K. C. Claffy and N. Brownlee, "Understanding Internet traffic streams: Dragonflies and Tortoises.," *IEEE Communications*, 2002.

[12] E. Kohler, J. Li, V. Paxson, and S. Shenker, "Observed structure of addresses in IP traffic.," in *Proc. ACM Internet Measurement Workshop*, 2002.

[13] K. Papagiannaki, N. Taft, S. Bhattacharyya, P. Thiran, K. Salamatian, and C. Diot, "A pragmatic definition of elephants in Internet backbone traffic.," in *Proc. ACM Internet Measurement Workshop*, 2002.

[14] K. C. Claffy, H.-W. Braun, and G. C. Polyzos, "A parameterizable methodology for Internet traffic flow profiling," *IEEE Journal on Selected Areas in Communications*, 1995.

[15] K. Thompson, G. J. Miller, and R. Wilder, "Wide-area Internet traffic patterns and characteristics," *IEEE Network Magazine*, 1997.

[16] S. Lin and N. McKeown, "A simulation study of IP switching," in *Proc. ACM SIGCOMM*, 1997.

[17] A. Feldmann, J. Rexford, and R. Caceres, "Efficient policies for carrying Web traffic over flow-switched networks," *IEEE/ACM Transactions on Networking*, 1998.

[18] P. Newman, G. Minshall, and T. Lyon, "IP switching: ATM under IP," *IEEE/ACM Transactions on Networking*, 1998.

[19] W. Stallings, *SNMP, SNMPv2, SNMPv3 and RMON 1 and 2*. Addison-Wesley, 1999.

[20] Cisco Netflow. `http://www.cisco.com/warp/public/732/netflow/index.html`.

[21] R. Sommer and A. Feldmann, "NetFlow: Information loss or win?," in *Proc. ACM Internet Measurement Workshop*, 2002.

[22] C. Barakat, P. Thiran, G. Iannaccone, C. Diot, and P. Owezarski, "A flow-based model for Internet backbone traffic," in *Proc. ACM Internet Measurement Workshop*, 2002.

APPENDIX

A. SOFTWARE COMPONENTS

Processing a very large number of flows into bins and aggregating and ranking them is a non-trivial exercise. Given the exploratory nature of the work, we decided to have an evolvable intermediate format that can help us answer questions quickly but retain enough information in order to look for alternate explanations. Note that different slices on the data require different parts of the flow records; e.g., per-application analysis requires port numbers and protocol information, while others require source and destination prefixes.

We divided the task into four components: restitching of the flows, binning, aggregation, and ranking. We used *unsampled* Netflow [20] records, ordering them temporally, and extracted the following fields: source and destination IP address/port number/AS number/prefix mask length, time fields, number of bytes/packets, protocol, and TCP flags. Netflow records usually come directly from a router. For the purpose of evaluation our software is capable of generating Netflow records as well as FragFlow records from detailed packet traces. Both constitute inputs for the binning step.

The base data structure used for the restitching of flows consists of a splay tree with orderings based on flow start time and a hash table using a flow key composed of source/destination IP address/port and protocol for fast flow information lookup. A flow can be accessed either by its key or the start time giving us the necessary flexibility for our analysis. Netflow records with the same key that overlap in time or follow each other within an inactivity timeout period (default 15 seconds) are merged into a single flow. In addition, associated information such as bytes and packets are summed, flags are `OR`ed, start time is set to be minimum of the two times etc. We process the flows using a sliding window data structure that covers all the active flow records while limiting the number of flows that need to be kept in memory. Really long flows hinder in moving the window forward. Thus, as a temporary backup procedure, we write to disk. In the future we will migrate to a merge-sort procedure to obviate the use of disk.

The binning phase uses the number of bytes contributed to the bin as a ranking key. The bins are ordered by bin start times. Raw flows are represented via a priority queue while aggregated flows are stored in a splay tree (since their ranks change) which contains all raw flows with the same aggregation key. Source (destination) prefix aggregation is done based on flows that share source (destination) prefixes with same source (destination) prefix mask lengths. If both source and destination prefixes match we call this prefix aggregation. We can also aggregate at a particular prefix mask length (e.g., 16, 24 etc.) or at source/destination AS level. We use a segment abstraction to partition aggregated flows using an inactivity timeout just as we do for raw flows. The segment abstraction aids in comparing duration of raw and aggregated flows.

Since all the bytes (packets) from a flow may not fall into complete bins we keep track of fractions of bytes (packets) that fall into incomplete bins and use average flow bandwidth to determine the fraction. The ranking of raw flows for each bin is done using standard priority queue insertion of flows with highest priority for ranks with the smallest number of bytes. This allows periodic culling of the flows with the least weight to limit the size of the priority queue. The aggregated flows are ranked by adding byte contributions to appropriate segments. Culling is harder since another flow belonging to an aggregated flow may arrive at some later time and with its contribution changes the priority of this aggregated flow. Their ranking is obtained by extracting flows in reverse order from the priority queue (ascending order of number of bytes). For aggregated flows, we remove excess flows while obtaining the ranking.

The resulting information is stored in a DBMS (PostgreSQL), which allows for highly flexible data handling. For example, we can derive more detailed information like the raw flows that are the contributors of an aggregated flow.

About 12,000 lines of C code forms the code base (5000 of which are shared as libraries for the entire system and the remaining ones are split roughly equally between restitching, binning, and flow collection). 1200 lines of shell, Perl and SQL scripts are used to load data into the DBMS and extract information from it. S-plus is used to generate the plots. The entire process is automated with a variety of supplied parameters, such as flow time out, binning-size, rank count, aggregation level, prefix mask length etc. Restitching takes the bulk of the time and once that is complete different bin sets can be constructed in parallel. Filters are specified as dynamically loadable shared library modules. Output is separated into a file for ranking and a flow lookup table, with the latter containing per-segment information for aggregated flows. Extending the software to handle a new aggregation type requires addition of a simple comparison function for looking up a matching aggregated flow.