2017 29th International Teletraffic Congress

Skampling for the Flow Duration Distribution

Nelson Antunes
Center for Computational and
Stochastic Mathematics, University of Lisbon
and University of Algarve
Email: nantunes@ualg.pt

Vladas Pipiras
Department of Statistics
and Operations Research
University of North Carolina
Email: pipiras@email.unc.edu

Darryl Veitch
School of Computing
and Communications
University of Technology Sydney
Email: Darryl.Veitch@uts.edu.au

***Abstract*—This paper concerns the problem of estimating the Internet flow duration distribution from indirect measurements due to network constraints. The aim is to estimate the distribution from observing: the possible superpositions (collisions) of sampled flow durations, the flow arrivals-to-departures times without identification of sampled flows and the number of sampled flows in progress. For each type of data available, we present estimators of the flow duration distribution, formulating the problem in queueing system terms. We also propose data streaming algorithms using sampling and sketching (through counters) to obtain the considered partial information from flows. At the core of this skampling (i.e. sampling and sketching) approach is the ability to tune the flow sampling probability for "optimal" flow load onto sketch entries (queues). Finally, we present numerical results comparing the different estimators of the flow duration distribution using two real Internet traces.**

***Index Terms*—Internet traffic measurements, packet and flow sampling, sketching, flow duration distribution, non-parametric estimation.**

I. Introduction

This paper concerns the measurement of flows of packets in Internet traffic, in particular of associated distributions such as those of flow sizes (number of packets in a flow) and durations (time interval from the first to the last packet). These distributions are important metrics in measuring and monitoring Internet traffic, each being used for traffic modeling, management, anomaly detection and accounting (see e.g. [1]). From the viewpoint of a user, the distribution of flow durations is arguably the more important one, and is the focus of this work.

Estimating such distributions poses unique challenges due to the computational and storage costs associated with the large volume of traffic. For example, a naive approach would be to keep a flow table, updating it with each passing packet (updating packet counts for flow sizes, and times for flow durations). However, such an approach is prohibitively expensive: each table entry must store a unique flow key which must be read and compared on each packet. This is especially slow when flows with long bursts of short closely-spaced packets arrive, each of which requires a flow key lookup.

As described in more detail in Sec. II, various packet sampling and sketching techniques have been suggested to overcome these challenges. In sampling, only a subset of all packets are selected, making "sampled" flows, which form the basis for inference. In sketching techniques, compact data structures with fast update rules are used to inexpensively, but only approximately, summarize information about the packet stream, the errors being accounted for in the inference.

Our work is inspired by a combined sampling and sketching approach for the flow *size* distribution, called skampling (a portmanteau of sketching and sampling), developed in [2]. The approach in [2] comprises three main components. First, sampling is performed at the flow level with probability p (all packets of a flow, or none, will pass). Second, sampled packets are hashed into an array of A counters based on a flow key, so all packets from a given sampled flow are mapped to the same counter. Third, after some measurement interval the packet counts are exported and used to estimate the flow size distribution. The key difficulty is that of flow collisions: a given counter may have counted packets from more than one flow, obliging the subsequent inference to perform deconvolution. The key advantage is that the flow sampling stage is performed free of flow table costs, and p can be chosen to control the collision rate, and thereby to optimize a tradeoff of "information destruction" caused by collisions against the volume of collected data. The end result is a computationally feasible method with a performance which is comparable to that of true flow sampling, which is in fact optimal.

In this work we propose a skampling approach for the flow *duration* distribution $D(t)$, which similarly to the case of flow sizes above, involves sampling, then sketching and "deconvolution". Exactly as in the flow size case, the first stage is a flow sampling step which acts on a flow key, which we assume can be determined for each packet. We also assume that the first, and for the moment also the last, packets of flows can be identified, the archetypal examples being the SYN and FIN packets of TCP connections which constitute the majority of traffic in the Internet. (The ideas and results of this paper apply to other types of flows provided that suitable substitutes can be found for connection startup and termination.) However, a crucial point is that because the sketch will not store flow keys, we cannot match up the SYN and FIN from the same flow j, so the flow duration *cannot be measured directly* as $t_{\mathrm{FIN}_j} - t_{\mathrm{SYN}_j}$. This being the case, how then can durations be accessed, and what form of sketch can capture this information?

To gain a handle on the problem we exploit a model of flow structure, known to be accurate for backbone links ([3],

978-0-9883-0453-6/17 $31.00 © 2017 ITC
DOI 10.1109/ITC.2017.27

[4]), whereby flows arrive as a Poisson process of intensity λ, with i.i.d. flow durations. Under these conditions, the dynamics of flow arrivals and departures is precisely mirrored by arrivals and departures to an $M/G/\infty$ queue, where the number of flows simultaneously active is just the number of active servers, commonly referred to as "queue length", and where the service or sojourn time distribution is simply the distribution $D(t)$ of flow durations. Flow sampling with probability p interacts in a very natural way with this model: the sampled flows can be modelled by the same $M/G/\infty$ queue with intensity reduced to $\lambda_p = \lambda p$, and the subset of these hashed to an entry in the sketch array of A elements corresponds to a queue with $\lambda_q = \lambda_p/A = \lambda p/A$.

The utility of the above model is that the literature on the $M/G/\infty$ queue documents many relationships between queueing observables, for example busy period durations, and the service time distribution $D(t)$ of interest here, which can form the basis of an inference method. Indeed, there has been a growing interest in recent years (e.g. [5], [6]) in statistical inference problems where partial measurements of queues can be used to infer various queueing parameters. Of the available approaches relevant to the estimation of service distributions, we select three where the required data is amenable to collection by simple counters acting on the raw packet arrival information, based respectively on busy periods (BP), queue length (QL), and "arrivals-to-departures" (AD) information.

It may not be possible to measure the equivalent of a FIN packet for some flow definitions, and it is well known that even many TCP flows do not terminate correctly. It is therefore important to see what can be done without FIN information. Of the above approaches, it is BP that shows the most promise. We accordingly also study a variant using an heuristic which can estimate BP information without the need for FINs. Note the inherent difficulty here: upon the arrival of a SYN packet, the decision has to be made on whether it starts a new busy period, or happens to fall within the duration of an existing flow (in which case the busy period continues). The detail of the approach requires going beyond the $M/G/\infty$ model to consider packet arrival structure within flows.

Because duration measurement involves timekeeping, the situation is naturally more complex than the case of flow size [2]. Whereas a single counter counting arrivals over a measurement interval was sufficient in that context, here an entry in the sketch array may involve multiple counters, and require state dependent actions rather than a simple increment. Moreover the need to measure time using counters implies that time must be discretized, and tracked asynchronously of packet arrival events.

Unless stated otherwise, we focus on a single sketch entry, fed by a stream of flows which we think of as associated to an $M/G/\infty$ queue of arrival intensity λ_q. We follow the approach of [2] by considering sketch entries to be mutually independent. Final sketch estimates can therefore be obtained by combining those from each entry in a straightforward way, effectively increasing sample size by a factor of A.

The proposed skampling approaches for the flow duration distribution are examined through simulation on two Internet traces, for which they show excellent performance. In particular, they outperform the sampling approaches of [7], [8], [9], [10], in that estimation is improved over the main body and the tail of the distribution, and in that they do not assume any special asymptotic structure as was the case in [8], [10]. Moreover, the performance is satisfactory for very small sampling probabilities for flows, for example as low as $p/A = 10^{-4}$, which cannot be handled by earlier approaches.

Our work can also be applied in other contexts such as stream processing and "big data", so long as the data consist of objects (packets) that are grouped into types (flows), for which there is an equivalent of a SYN packet. In stream processing, the objects cannot be stored and must be analyzed on the fly, and in the context of "big data", fast data summaries are needed. The corresponding metric of interest consists of the frequencies of the durations of object types.

The paper is organized as follows. We review briefly the literature in Sec. II. In Sec. III, we establish the sampling framework based on the $M/G/\infty$ queue, and derive the estimators for the three types of partial information considered. The sketching algorithms to obtain the considered partial information are presented in Sec. IV. The results are then applied to two Internet traces in Sec. V and, finally, conclusions and future directions are presented in Sec. VI.

II. Related Work

Sketches have been proposed to measure many metrics of interest including heavy hitters [11], super-spreaders [12], frequency moments [13], entropy [14], flow size distribution [15], and inverse distribution [16]. Very few works in this area concern flow durations, with the exception of [17] and [18] to the best of our knowledge. In [18], working with older traces from backbone links, all flow durations are tracked using a set of Bloom filters, each representing flows falling in a certain range of bin durations. To track the durations of long flows, an aging process is used. In [17], the interest lies in characterizing long-lived flows with a dedicated efficient data streaming algorithm. In both of these works, there is no statistical estimation involved in the sense that the estimates are obtained directly from the output of the complex sketching algorithms.

On the sampling side, most of the focus has been on the estimation of the flow size distribution [19], [20]. Assuming a Bernoulli sampling of packets, the flow duration distribution is estimated for large flows assuming a particular form of the distribution in [21]. The works [7], [8], [9], [10] have proposed an analytical framework under several methods of sampling packets to estimate the flow duration distribution. This approach requires modeling the duration distribution in terms of the interarrival times (IATs) between packets and flow size. The inversion of flow distributions (IATs and size) from sampled flow quantities allows the flow duration distribution to be estimated, but the procedure is prone to numerical issues associated with the inversion.

Our work also involves sampling and inference (from sketches) but is devoid of some of the earlier difficulties. First, the sampling is at the level of flows, as opposed to packets. Second, as a result, the considered inference procedures show superior performance at suitable sampling rates, which can be tuned at the sampling stage. These ideas have been suggested explicitly and studied in [2] for the distribution of flow sizes.

III. Inference Analysis

In this section we deal with the queueing-theoretical underpinning of the inference methods, essentially how the flow duration distribution can be expressed in terms of *accessible* arrival and departure information.

As described above, we suppose that sampled flows arrive (to a sketch entry) in a Poisson stream at rate λ_q, and their durations are assumed to be independent of each other and of the flow arrival process. Therefore, the arrivals and departures of flows can be formulated as an $M/G/\infty$ queue model which has been used extensively in the networking literature. This is a semi-parametric model with arrival intensity λ_q and the cumulative distribution function (CDF) $D(t)$ characterizing the system. In a slight abuse of notation, we will use the same letter, D in this case, to denote the corresponding random variable. The queue analysis is tractable. For example it is well known that the number of flows in progress in equilibrium is Poisson distributed with mean $\lambda_q E[D]$.

We consider three types of accessible partial information:

(i) the indicator busy period process $(\mathbf{1}_{\{Q(t)>0\}})$, where $Q(t)$ is the queue length at time t, indicating whether the system is in a busy or idle period;
(ii) the arrivals-to-departures information, consisting of times from flow departures to their most recent flow arrival;
(iii) the queue-length process $(Q(t))_{t\geq 0}$ in equilibrium over a finite time interval.

The different partial information will be abbreviated as: **BP** (Busy Periods), **AD** (Arrivals-to-Departures) and **QL** (Queue Length). Note that the data of type QL is more informative than those of types BP and AD, in the sense that both types BP and AD can be reconstructed from the data of type QL, but not vice versa. The BP and AD types cannot be ordered, in the sense that neither can be reconstructed from the other. The three types of partial information are considered separately next. Different inference procedures will be used depending on the partial information. These will be compared in Sec. V.

BP information. The aim here is to estimate the distribution $D(t)$ of flow durations using only the information of the idle periods and busy periods of the queue. It is known (see e.g. [22]) that the Laplace-Stieltjes transform (LST) of the distribution $B(t)$ of busy periods depends on λ_q and $D(t)$ through the formula

$$\widetilde{B}(s) := \int_0^\infty e^{-sx} dB(x) = \widetilde{I}(s)^{-1} - \left(\lambda_q \int_0^\infty e^{-st} e^{-\lambda_q \int_0^t (1-D(x))dx} dt\right)^{-1}, \quad (1)$$

where $\widetilde{I}(s) = \lambda_q/(\lambda_q + s)$ is the LST of the distribution of an idle period which is exponential with parameter λ_q. The mean busy period derived from Eq. (1) gives

$$E[B] = (e^{\lambda_q E[D]} - 1)/\lambda_q, \quad (2)$$

which is finite if and only if the flow duration has finite mean. From Eq. (1) we can obtain the distribution $D(t)$ which was noted in [23] and also investigated in [24] for the $M/G/\infty$ queue. More specifically, for two functions $F(t)$ and $G(t)$, $t > 0$, define their convolution as the function $(F*G)(t) = \int_0^t F(t-x)dG(x)$, $t > 0$. Let $C(t) = (B*I)(t)$ be the distribution of a busy cycle (the sum of a busy period and an idle period). The renewal function associated with the distribution $C(t)$ is $U(t) = \sum_{k=0}^\infty C^{*k}(t)$, where $C^{*0}(t) = \mathbf{1}_{\{t\geq 0\}}$ and $C^{*k}(t)$, $k \geq 1$, is the k-fold convolution of $C(t)$. Since $\widetilde{C}(s) = \widetilde{B}(s)\widetilde{I}(s)$, it follows from Eq. (1) that

$$\lambda_q \int_0^\infty e^{-st} e^{-\lambda_q \int_0^t (1-D(x))dx} dt = \widetilde{I}(s)(1 - \widetilde{C}(s))^{-1} \quad (3)$$

and therefore

$$\lambda_q e^{-\lambda_q \int_0^t (1-D(x))dx} = (I * U)(t). \quad (4)$$

Solving with respect to $D(t)$, we obtain

$$D(t) = \frac{U'(t)}{(I' * U)(t)}, \quad t > 0, \quad (5)$$

where $U'(t) = \sum_{k=1}^\infty (I'*B)^{*k}(t)$ is the density of the renewal function and $I'(t)$ represents the exponential density function with parameter λ_q.

Given a measurement window with n idle periods $I_1, I_2, \ldots, I_n$ and busy periods $B_1, B_2, \ldots, B_n$, we can estimate the sampling rate and busy period distribution as

$$\widehat{\lambda}_q = \Big(\frac{1}{n}\sum_{i=1}^n I_i\Big)^{-1}, \quad \widehat{B}(t) = \frac{1}{n}\sum_{i=1}^n \mathbf{1}_{\{B_i \leq t\}}. \quad (6)$$

Then, an estimator for $D(t)$ is

$$\widehat{D}(t) = \frac{\widehat{U}'(t)}{(\widehat{I}' * \widehat{U})(t)}, \quad (7)$$

where $\widehat{U}(t)$ and $\widehat{I}(t)$ are the empirical counterparts of $U(t)$ and $I(t)$ obtained by using (6). Note that an estimate for $E[D]$ can be obtained directly through (2) by using the empirical counterparts of λ_q and $E[B]$. See [23] for more theoretical properties of the estimator $\widehat{D}(t)$.

The effect of the magnitude of λ_q should also be noted here, since it is a parameter we can control. For λ_q close to 0, busy periods are very likely to be trivial, consisting of a single flow, and with $B = D$. The estimator $\widehat{D}(t)$ will reflect this, though this is not immediately evident from the expression (5). The downside is that there will be very few busy period samples available. As λ_q increases, busy periods will first become increasingly frequent, but will also begin to become non-trivial ($B > D$), making inference harder. As λ_q increases further they will begin to merge and so become fewer in number, ultimately just a single continuous busy period,

thus starving BP-based methods of data. Without sampling (i.e. $\lambda_q = \lambda$), this is the case under realistic network conditions, where the queue will have 10's or even 100's of thousands of concurrent flows. Control via sampling and hence smaller λ_q is then essential to achieve a regime where there are a sufficient number of BPs which are, intuitively, equal or close to durations.

AD information. The partial information here consists of the time intervals between flow departures and the immediately preceding flow arrivals, referred to as the arrival-to-departure times.

Let Y_i be the time of the ith departure, D_i the duration of the flow ending with the ith departure, and Z_i be the time interval between Y_i and the latest arrival preceding it, that is, the arrival-to-departure time. Brown [25] showed that the CDF of Z_i given $D_i = d$ can be expressed as $Z(z|d) = (1 - e^{-\lambda_q z})\mathbf{1}_{\{z<d\}}$. Integrating over the values of d (see [25] for details), the CDFs $Z(t)$ and $D(t)$ can then be related as $Z(t) = 1 - (1 - D(t)))e^{-\lambda_q t}$, which leads to

$$D(t) = 1 - (1 - Z(t))e^{\lambda_q t}. \tag{8}$$

Supposing that n departures and hence variables Z_i are observed, we can estimate the arrival rate $\widehat{\lambda}_q$ from the arrival times of the sampled flows and the CDF $Z(t)$ by $\widehat{Z}(t) = n^{-1}\sum_{i=1}^{n}\mathbf{1}_{\{Z_i\le t\}}$. By replacing these quantities in Eq. (8), we obtain an estimator for $D(t)$,

$$\widehat{D}_0(t) = 1 - (1 - \widehat{Z}(t))e^{\widehat{\lambda}_q t}. \tag{9}$$

Since $\widehat{D}_0(t)$ estimates a CDF but is not necessarily a non-decreasing function, the following final estimator is taken for $D(t)$:

$$\widehat{D}(t) = \sup_{0\le u\le t}\widehat{D}_0(u). \tag{10}$$

Theoretical properties of the estimator $\widehat{D}(t)$ are considered in [25], and an extension to the times between departures and and the rth preceding arrivals in [5].

As in the BP case, λ_q plays an important role. For large λ_q, the duration of flows D_i are relatively large compared to the flow interarrival times, and therefore Z_i tend to be smaller. In view of (9), $\widehat{Z}(t)$ and hence $\widehat{D}_0(t)$ would tend to reach 1 more quickly (smaller t), and so the estimation of $D(t)$ for larger t becomes more problematic. On the other hand for small λ_q, $\widehat{D}_0(t)$ is essentially $\widehat{Z}(t)$ by (9) but there will be very few, if any, Z_i samples available.

QL information. The last partial information considered in order to estimate $D(t)$ is the queue length process. The auto-covariance function of the process $(Q(t))_{t\ge 0}$ is

$$R(t) := \mathrm{Cov}(Q(s), Q(s+t)) = \lambda_q\int_t^\infty (1 - D(x))dx \tag{11}$$

(see e.g. [23], [6]). Differentiation yields

$$1 - D(t) = -\frac{1}{\lambda_q}R'(t), \quad t > 0. \tag{12}$$

This relationship provides the basis for the construction of an estimator of $D(t)$. Supposing the observation time interval $t \in [0, T]$ is discretized as ih, $i = 1, \ldots, n$, with $h > 0$, consider the sample mean $\overline{Q} = \frac{1}{n}\sum_{i=1}^{n} Q(ih)$, and let

$$\widehat{R}(jh) = \frac{1}{n}\sum_{i=1}^{n-j}(Q(ih) - \overline{Q})(Q((i+j)h) - \overline{Q}) \tag{13}$$

be the sample auto-covariance of Q at time lag jh. The estimator of $D(t)$ at time $t = jh$, $j = 1, \ldots, n$, is given by

$$\widehat{D}(jh) = 1 + \frac{1}{\widehat{\lambda}_q}\frac{\widehat{R}(jh) - \widehat{R}((j-1)h)}{h}, \tag{14}$$

where $\widehat{\lambda}_q$ can be estimated from the up-jump times of the queue length process associated with flow arrivals. See [23] for properties of some estimators related to $\widehat{D}(jh)$.

Like the BP and AD cases above, a small λ_q translates to a lack of data, but unlike them, large λ_q is not detrimental to estimation. On the contrary, since the covariance function depends on λ_q only via a prefactor, higher values simply means more queueing events, and so the estimator variance should monotonically improve with λ_q. Note however that this does not imply vanishing bias in the limit or consistency, since the measurement interval is finite and so the far tail of $D(t)$ is not sampled. This is borne out in our simulations reported below.

IV. Sketching Algorithms and Implementation

In this section we describe how the partial information considered in Sec. III can be collected through appropriate choices of sketch data structure/algorithm, acting on the packets from the input flows. As before we consider these to be those sampled flows directed to a single entry in the sketch array.

We measure time in discrete units of width Δ seconds using a counter, C_t, which is incremented at the end of each time slot. This counter is special as it is accessed in two ways: by the periodic increment just described, and independently by the sketch data structure logic as described below, which reacts to packet arrival events, which can occur at any time (not slotted). For simplicity, we assume that packets do not arrive precisely on slot boundaries.

We present results in the order AD, QL, and BP, of increasing sketching complexity. We then define a fourth approach, a BP based heuristic which does not require FIN packets. In each case the algorithms described run over the course of some observation window, exporting values as they go. At the end of the window these values are used to form estimates of $D(t)$.

AD information. A sketching algorithm for the AD information Z_i from Sec. III can be designed using only the C_t counter, defined above. Only SYN and FIN packets are used, others are ignored. The scheme works as follows, initialized at the arrival of the first SYN packet. Each time a SYN packet is seen, C_t is reset to zero. Whenever a FIN packet arrives, the current value of the counter is exported. It should not be forgotten here that C_t is also being incremented periodically in the background at the end of each time slot.

QL information. A sketching algorithm to collect the QL information can be implemented using two counters C and C_t,

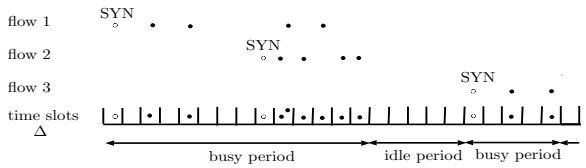


Fig. 1: Construction of busy and idle periods from flows.

representing respectively the queue length, and the number of time slots since the last change of the queue state. Again only SYN and FIN packets are used, others are ignored.

The scheme works as follows. At the beginning of the observation window we assume the queue is empty and initialize $(C, C_t) = (0, 0)$. Each time that a SYN or FIN arrives, C is incremented or decremented, respectively, and at the end of such a time slot (C, C_t) is exported (before the periodic C_t increment) and $C_t = 0$ is reset (just after the periodic increment).

It is possible that C can be negative due to the arrival of FIN packets from flows that are active at the beginning of the observation window. In this case we subtract the minimum of the exported C values from those values so the new minimum is zero. It is still possible however that the queue was not initially empty but this is not detected, in which case the exported QL values will be too low by some positive integer.

From all the exported pairs (C, C_t), the queue length process can be reconstructed straightforwardly.

BP information. Busy period measurements can be obtained using the same values exported from the QL algorithm above, an approach we refer to as the *BP via QL*.

Inferred BP information. We now define the *inferred BP algorithm*, which uses an heuristic to estimate busy period durations without using FIN packets.

To explain the basic idea, note that a busy period always starts with a SYN packet (see flows 1 and 3 in Fig. 1) but that, due to flow "collisions", a SYN packet may fall within the duration of another flow and hence may not initiate a busy period (flows 1 and 2 in Fig. 1). Thus, the time between a SYN packet and the previous packet is either an idle period (in which case the SYN packet starts a busy period) or a (proper) subset of an interarrival time (IAT) between two consecutive packets of another flow.

Recall from Sec. III that an idle period is exponentially distributed with parameter λ_q and hence has mean $1/\lambda_q$. The key insight is that in our skampling framework λ_q can be chosen to be small enough so that an idle period will tend to be larger than the IATs between consecutive packets of a single flow, and hence even larger than the IATs within a busy period with many flows. In such a regime whether a SYN packet starts a busy period or not can be inferred by comparing the interval between the SYN and the previous packet to a threshold value Γ, based on some quantile of the idle period distribution.

In fact, we make Γ an increasing function of the number of packets in the current (inferred) busy period to make idle detection progressively harder to achieve, in order to reduce the risk of a false positive, that is concluding that the SYN begins a new busy period when this is not the case. The motivation is twofold: (i) burstiness of in-flow IATs means that opportunities for false positives naturally grow with flow duration; (ii) with reference to the well-known mouse/elephant dichotomy for flows, a long busy period due to sampling an elephant will be characterized by the higher IAT variability of such flows. Details on the choice of Γ are left to Sec. V.

Algorithm: Inferred BP information

Initialization: First time packet arrives in obs. window:
set $C_t = 0$, $C_d = 0$, and $C = 1$;

Loop: Upon packet arrival:

Case 1a: If $0 \le C_t \le \Gamma(C)$, or

Case 1b: $C_t > \Gamma(C)$ and not a SYN: `% BP continues`
set $C_d = C_d + C_t$; $C_t = 0$; $C = C + 1$;

Case 2: If $C_t > \Gamma(C)$ and a SYN `% new BP`
export C_d; set $C_t = 0$; $C_d = 0$; $C = 1$;

We now turn to the description of the inferred BP algorithm. Three counters are needed:

- C_t measures the intervals between consecutive packet arrivals ($C_t - 1$ holds the number of slots since the last arrival);
- C_d holds the duration of the current busy period (units of Δ);
- C holds the current size (# of packets) of the busy period.

Finally, $\Gamma = \Gamma(C)$ denotes the idle detection threshold described above. Some of the cases in the algorithm above can be illustrated with the flows represented in Fig. 1. The arrival of flow 2 corresponds to Case 1a and the arrival of flow 3 to Case 2 (assuming the conditions involving $\Gamma(C)$ are satisfied). The arrival of the second packet in the same time slot (tenth slot in Fig. 1) corresponds to Case 1a. The algorithm incorporates some ideas of [17] for tracking durations of long flows, where several counters with fast update rules and exporting are used.

V. Data study

In this section, we will assess the quality of and compare the proposed estimators using real Internet traffic data. We consider two publicly available Internet traces, Auckland IX and Waikato V.[1] A summary of some of the trace statistics is given in Table I. We assume that estimates of λ and the mean flow load factor $\rho = \lambda E[D]$ are available. The arrival rate λ can be estimated by counting SYN packets, and ρ can be estimated over some preliminary short measurement interval. The quality of the estimators will be evaluated as a function of the mean load factor $\rho_q = \lambda_q E[D]$ representing the level of superposition (collision) of flows in the queue, which is strongly tied to inference performance.

We first consider the ideal case in which the SYN and FIN packets of flows can be observed (removing the flows with one packet) from which the BP, AD and QL partial information can be reconstructed in order to have a fair comparison across

[1] http://wand.net.nz/wits

Trace	Duration	# Packets	# TCP Flows	λ (flows/s)	$E[D]$(s)
Auck. IX	1h	38,308,012	1,371,756	346.8	11.0
Waik. V	1h25m	15,486,413	842,578	156.1	7.6

TABLE I: Traces summary.

the different methods with respect to the ground truth. The existence of a small percentage of flows with only one packet (SYN) or duplicate FIN packets could perturb momentarily the quantities exported in the AD algorithm but will have negligible impact in the estimation. For the QL information the impact could be mitigated with additional changes similar to the ones described at the end of the algorithm. On the other hand, the inferred BP algorithm is more robust to these kinds of flows. We give results for a single sketch entry and, in addition to comparison across different methods, are interested in the values of ρ_q for "optimal" performance. At the end of the section, we consider the estimation of the flow duration distribution with the inferred BP information (see Sec. IV).

The choice of time slot width Δ impacts on the granularity and accuracy of the data captured in the sketch, and on the subsequent estimation. It is constrained in practice by measurement infrastructure in terms of available processing time and memory. For our off-line analysis here we use a value of $\Delta = 10^{-3}$s, or 1 millisecond, in all cases.

Model adequacy. The assumption of homogeneous Poisson flow arrivals was examined for both traces through testing the uniformity of the arrival time distribution, conditionally on the number of arrivals. It passed standard statistical tests (e.g. Kolmogorov-Smirnov). The other assumptions involved were the independence of flow arrival times and duration times, and that duration times form a sequence of i.i.d. random variables. We did not test these here, but if there are reasons to suspect violations, statistical tests are available, as described in [26].

BP information via QL. Fig. 2a concerns the estimation of the (complementary) flow duration distribution for several values of ρ_q for the Auckland trace, using the BP via QL skampling scheme. For the considered values of ρ_q, the combined sampling probability p/A ($= p$ since $A = 1$ sketch entry is considered) is smaller than 10^{-4}.

For each value of ρ_q, we obtain 100 independent replications of our $D(t)$ estimates through randomly resampling the trace data input to the sketch entry. We use these to empirically measure the distributional properties of our estimators based on a **single** sketch entry. This should not be confused with using a sketch with $A = 100$ entries, where the flow samples are generated via a hash function and so are not perfectly independent across the sketch entries.

The plots show the median of 100 empirical (complementary) CDFs. In other words, we plot the median of the 100 estimates at each value of t with time truncated at 200 seconds, which covers more than 0.99 of the probability (i.e., $D(200) > 0.99$). The performance of the estimator in the tail is better when ρ_q takes values between 0.5 and 1.5. The main body of the distribution is estimated well over the full range shown, as also seen from the small insert on the bottom left of Fig. 2a.

As a criterion for quantifying the quality of the estimator, for each replication we use the average difference $(1/T)\int_0^T |D(t) - \widehat{D}(t)|\, dt$ with $T = 200$. Fig. 3a shows boxplots (constructed using the 100 replications) of the average differences for the same ρ_q range. The tradeoff with respect to ρ_q discussed earlier is seen here. When ρ_q is equal to 0.1, the inversion difficulty due to collisions is smaller as busy periods correspond essentially to the sampled flow durations; however, the number of busy periods obtained is also smaller, increasing the variability in the estimation (here, $E[D] = 11$ from Table I, $E[B] = 11.6$, $E[I] = 110.3$, and dividing the length of the trace by the mean busy cycle $E[B] + E[I]$ gives 29.5 samples of busy periods on average). When $\rho_q = 2$, we see the worst result since there is both many more collisions in busy periods, and fewer than "optimal" (see $\rho_q = 1$ below) sampled busy periods available (here, $E[B] = 35.3$, $E[I] = 5.5$, and 88.2 samples of busy periods on average).

The outliers in the boxplots correspond to extreme cases due to the sampling of very long flow durations which decrease the sample size markedly. This is more likely as ρ_q increases. The insert on the bottom left of Fig. 3a plots the medians for a wider range of values ρ_q and fits a curve to them. We can see that the smallest average difference is attained around $\rho_q = 1$ (here, $E[B] = 19$, $E[I] = 11$, and 120 samples of busy periods on average).

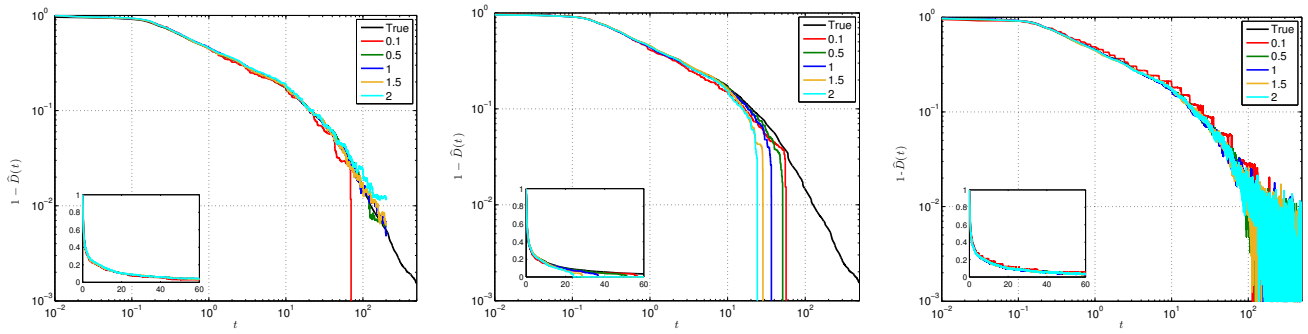
The "optimal" value $\rho_q = 1$ should perhaps not be surprising. The performance of the estimator depends not only on the load factor ρ_q but also on the number of sampled busy periods in the estimation. For a fixed p, the mean busy cycle is $(e^{\lambda_q E[D]} - 1)/\lambda_q + 1/\lambda_q$ by using (2). The minimum value of the mean busy cycle is attained when $\lambda_q = 1/E[D]$, yielding $\rho_q = 1$ and $p/A = 1/(\lambda E[D])$. At this value the number of busy period samples is maximized, which is one of the key factors controlling the quality of the estimation.

Figs. 4a and 4b depict the estimation of the flow duration distribution and the boxplot of the average differences for the Waikato trace, respectively, under the same settings used for the Auckland data. From the distribution function, we see that flows have smaller durations. The same behaviour of the estimator is observed and similar conclusions can be drawn.

AD information. Fig. 2b presents the medians of 100 estimates of $\widehat{D}(t)$ for several values of ρ_q as above, but using the Brown estimator with the Auckland trace (the vertical lines observed are due to $1 - \widehat{D}(t)$ being zero).

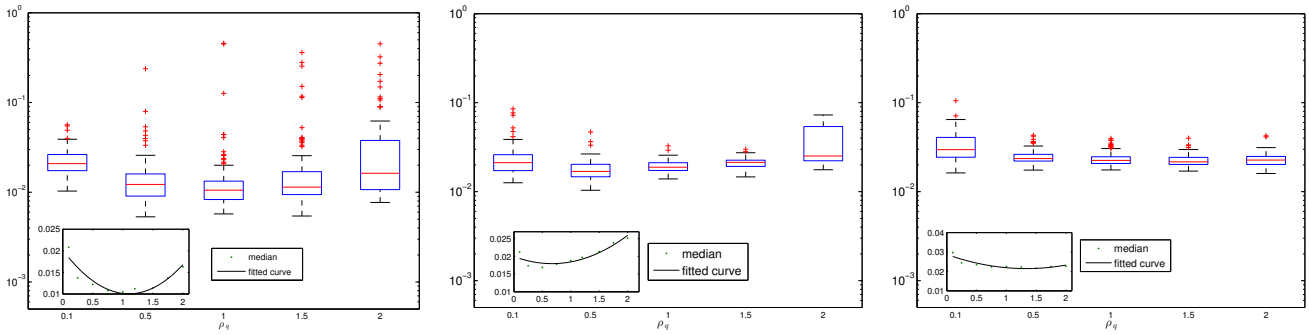
When $\rho_q = 0.1$, more values in the distribution tail are estimated but with more variability than in the main body of the distribution. The former is due to the following: as ρ_q increases, the durations of flows tend to be larger compared to the interarrival times between sampled flows, and so inference regarding $D(t)$ for large t becomes unreliable (see a related discussion in Sec. III). The generalization of the Brown estimator derived in [5] could be considered to improve the estimation in the tail as mentioned in Sec. III. However, we will not pursue this here, since the partial information uses the FIN packets and can be viewed as problematic.

The respective boxplots of the average differences between the true and estimated distribution functions over $[0, 200]$ are



(a) BP information via QL (b) AD information (c) QL information

Fig. 2: Flow duration distribution estimation for $\rho_q = 0.1, 0.5, 1, 1.5, 2$ – Auckland.



(a) BP information via QL (b) AD information (c) QL information

Fig. 3: Average differences between $D(t)$ and $\widehat{D}(t)$ over $(0, 200)$ – Auckland.

shown in Fig. 3b. The largest differences are attained at $\rho_q = 0.1$ because of the few samples of Z_i obtained and at ρ_q equal to 2 for the reasons given above. In order to describe the shape of the differences, the insert on the bottom left shows the median average differences for several values of ρ_q. We see that the minimum difference is around $\rho_q = 0.5$. Comparing Figs. 2a–3a with 2b–3b, we conclude that estimation based on the AD information is worse than that of the BP information.

We omit the results for the Waikato trace since there are no significant differences in the plots and similar conclusions can be drawn.

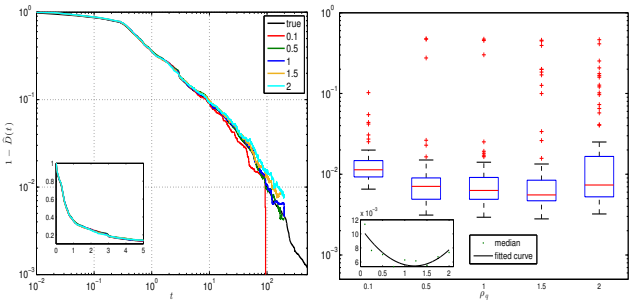
QL information. Finally, the performance of the estimator in Eq. (14) is shown in Fig. 2c using the respective algorithm in the same setting. From the figure, we do not see much difference in the medians of 100 estimates for the values of ρ_q considered, with the exception of $\rho_q = 0.1$, where the piecewise linear behaviour is due to the small changes of the queue length over time. However, there is now more variability in the estimation of the tail compared with the other approaches (cf. Figs. 2a, 2b and 4a). The boxplots of the average differences in Fig. 3c also confirm that estimation does not depend much on ρ_q for the range of the values considered, excluding $\rho_q = 0.1$. Comparing with the estimation provided by the BP and QL partial information, the median of the average difference in Fig. 3c is larger in the region between $\rho_q = 0.5$ and $\rho_q = 1$. The same observations apply to the Waikato trace, and we omit the figures for brevity.

Inferred BP information. Finally, we consider estimation of the distribution of flow durations from the inferred BP information. To define the function Γ of the algorithm described in Sec. IV that sets the threshold value for the beginning of a new busy period upon the arrival of a SYN packet, the well-known mouse/elephant dichotomy for flows will be used (see e.g. [21]). The following simplified definition will be used: a mice (resp. elephant) is a flow with the size less than or equal to (resp. more than) 30 packets. This definition may appear a bit crude at first glance, but provides a functional separation between long and short flows.

Indeed, these two types of flows are expected to have different behaviours in the network. The flows with less than 30 packets operate mostly in the slow start period of the TCP protocol. Long flows, on the other hand, are expected to be regulated by the congestion avoidance regime of the TCP protocol. When inspecting the packet transmission pattern of elephant flows in the traces and as also reported in the literature, one can observe that it is not regular but rather characterized by periods of low transmission rates. In our skampling framework we can control λ_q to be low, a long busy period is then often due to sampling an elephant.

As discussed in Sec. IV, if Γ increases with the number of packets in the current busy period, the arrival of a SYN packet in a large IAT between packets of an elephant is less likely to be classified as a new busy period.

We have tried several forms for the function Γ with an exponential growth, settling on the function given by $\Gamma(x) = \alpha + \beta e^{\gamma \min(x,30)}$, $x \geq 1$, which depends on positive parameters



(a) (b)

Fig. 4: BP information via QL (Waikato): (a) Flow duration distribution estimation for $\rho_p = 0.1, 0.5, 1, 1.5, 2$ and (b) Average differences between $D(t)$ and $\widehat{D}(t)$ over $[0, 200]$.

α, β and γ. Short busy periods are composed of (possible) superposition of mice flows, where the time between a SYN packet and the previous packet inside a busy period tends to be small compared with an idle period. Therefore, we set α as e.g. the 10th percentile of the idle period distribution (i.e. $I(\alpha) = 0.1$). As more and more packets are seen in a busy period, this indicates that an elephant flow has been sampled and Γ should increase as explained above. To fit β and γ, we specify two values of the function $\Gamma(x)$ at $x = 20$ and $x = 30$ packets, so that $\Gamma(x)$ grows slowly until 20 and then quickly up to 30 packets. The values $\Gamma(20)$ and $\Gamma(30)$ are taken as the 25th and 50th percentiles of the idle period distribution, respectively. The choice of $\Gamma(30)$ as the median $(\ln 2)/\lambda_q$ of the idle period counters the possibility of IATs between packets of an elephant flow being of the same order as the idle times (timeout values 15–60 sec). This is what could be considered as the extreme end of the parameter setting where the inferred BP algorithm has any chance to continue working: if many IATs are an order larger than the idle times, no algorithm can be expected to identify BPs even approximately. Along similar lines, the selected form of the threshold Γ should be critical in the cases closer to the "extreme" and conversely, matter little for large idle periods (corresponding to smaller p and ρ_q).

Fig. 5 depicts the duration distribution estimates when using the inferred BP information. In contrast to Figs. 2–4 where the results were given for a single sketch entry (queue), the results now use the busy periods from all A sketch entries, each with a given ρ_q. We implement the entries using random sampling however, rather than hashing. We use $A = 10, 20, 30$.

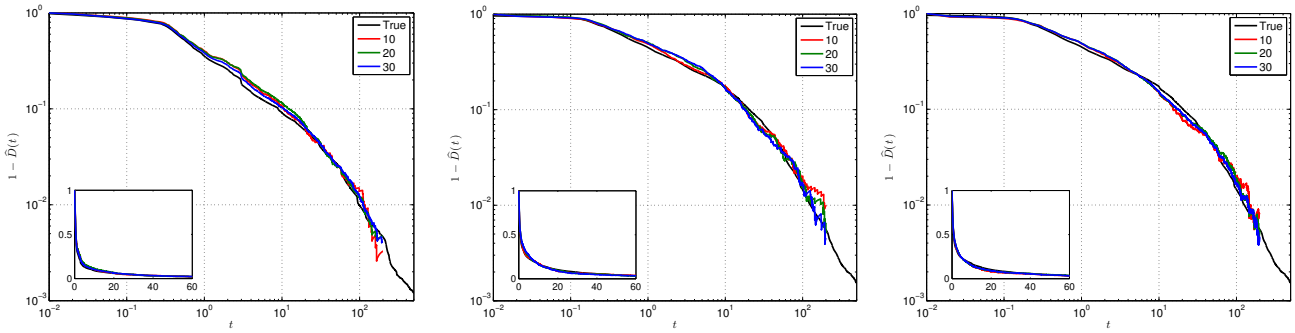
For the Waikato trace, the mean idle period $E[I]$ for ρ_q equal to 0.25 is 30 seconds, while for Auckland, the mean idle period for ρ_q equal to 0.5 and 1 are 22 and 11 seconds, respectively. A larger mean idle period allows for a more accurate decision when exporting the inferred busy periods. For Waikato, the total number of true busy periods (with $A = 30$) is 3810 and in 88.4% of them, the start of the the busy period were correctly detected. In the case of Auckland, with $\rho_q = 0.5$ (resp. $\rho_q = 1.0$) the total number of true busy periods is 2296 (resp. 3748) from which 82.5% (resp. 73.7%) were correctly detected. (Note that the total number of true busy periods is larger for Waikato due to the longer trace duration.) We can always increase the mean of the idle period $(1/\lambda_q)$ by choosing a smaller p for a better detection of the start of a busy period and compensating the decrease of the number of busy periods by using larger A, though with an increased operational cost. This is also related to the comparison with a collision free approach, where only one flow arrives per sketch entry. For Auckland, $A = 9785$ is needed to avoid collisions on average for the same sampling probability $p \approx 0.01$ corresponding to $\rho_q = 1$ and $A = 30$. However, in the former case there is no cost of exporting the information of the counters. The underlying optimization problem will be part of a future research; see below. Finally, we note that the main body of the distribution is well estimated for both traces as seen from Figs. 5a–5c.

VI. Conclusions and Future Work

In this work, we proposed a skampling (a hybrid of sampling and sketching) approach to estimate the distribution of flow durations on highly aggregated packet traffic links, nominally the carriage of TCP connections in the Internet.

If we think of a sketch as an array of counters or sets of counters into which flows are mapped via a hash function, it is not immediately clear how durations can be measured, given that counts will mix together the packets of many overlapping flows. We overcome this "collision" problem in two ways, first, by modelling the flow arrival process by an $M/G/\infty$ queue, and second, by using a flow sampling pre-filtering step. The queue based modelling allowed the considerable body of work in the queueing theory literature to provide options for inference approaches capable of "deconvolving" the collision induced non-linearities in the counter data. The flow sampling allows the flow collision rate to be reduced to a level where these methods can perform well, yet without the need for a costly flow table (which avoids collisions by indexing strictly via flow keys).

Three inference approaches were selected based on queueing theoretic relations, yielding sketching approaches involving either one, two, or three packet counters (note no retention of flow key), which could reconstruct exploitable queue quantities, or "partial queue information", from the base packet counter measurements. The best performing technique was based on the busy periods of the queue. Assuming the availability of both SYN and FIN packets of sampled flows, the busy periods were reconstructed exactly from the queue length process, which was tracked with two counters, one recording the queue length and the other recording times between its changes. Because FIN packets (or their equivalent under other flow definitions) are not always available, we also proposed a modified heuristic based on the busy period technique which does not require them. It involved three counters, of which two track discretised time intervals, and the other counts the busy period size (number of packets). The counters are incremented and/or reset according to temporal or packet arrival triggers. The proposed skampling methods were assessed on two real



(a) $\rho_q = 0.25$ – Waikato (b) $\rho_q = 0.5$ – Auckland (c) $\rho_q = 1$ – Auckland

Fig. 5: Flow duration distribution estimation - inferred BP information with several queues whose number is indicated in the legend.

Internet traces, showing excellent performance even for very small sampling probabilities.

Several questions related to this work could be pursued in the future. In terms of improving the current busy period based heuristic approach, one could investigate the case where some flows do have a FIN packet available which can be exploited, and others not. A better understanding of the choice of the threshold function $\Gamma(x)$ would be desirable as well. More broadly, other queueing relations could be considered, such as aggregation of the busy periods of the queue process over an observation window, reducing the frequency at which counts would have to be exported from the sketch. Another important question concerns a full analysis of how to optimize performance subject to some cost metric (such as total memory use for fixed estimation variance) within the proposed skampling approach. In this work, the flow sampling probability was chosen so as to optimize the load factor ρ_q per sketch entry, or "queue", with respect to the number of samples available, however there are other parameters at play which impact both on estimation quality and implementation cost. For example, increasing the size A of the sketch array improves estimation but increases memory costs, and more powerful queueing approaches may involve more counters per sketch entry, increasing statistical performance, but again at the cost of more expensive memory.

REFERENCES

[1] E. Cohen, N. Duffield, H. Kaplan, C. Lund, and M. Thorup, "Algorithms and estimators for summarization of unaggregated data streams," *J. Comput. Syst. Sci.*, vol. 80, no. 7, pp. 1214 – 1244, 2014.

[2] D. Veitch and P. Tune, "Optimal skampling for the flow size distribution," *IEEE Trans. Info. Theory*, vol. 61, no. 6, pp. 3075–3099, June 2015.

[3] N. Hohn, D. Veitch, and P. Abry, "Cluster processes: a natural language for network traffic," *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2229–2244, Aug. 2003.

[4] N. B. Azzouna, F. Clérot, C. Fricker, and F. Guillemin, "A flow-based approach to modeling ADSL traffic on an IP backbone link," *Ann. Telecommun.*, vol. 59, no. 11-12, pp. 1260–1299, 2004.

[5] N. Blanghaps, Y. Nov, and G. Weiss, "Sojourn time estimation in an $M/G/\infty$ queue with partial information," *J. Appl. Probab.*, vol. 50, no. 4, pp. 1044–1056, 12 2013.

[6] A. Goldenschluger, "Nonparametric estimation of the service time distribution in the $M/G/\infty$ queue," *Adv. in Appl. Probab.*, vol. 48, no. 4, pp. 1117–1138, 12 2016.

[7] N. Antunes and V. Pipiras, "Inverting flow durations from sampled traffic," in *Proc. ITC*, 2012, pp. 7:1–7:8.

[8] ——, "Estimation of flow distributions tails from sampled traffic," in *Proc. IEEE SSP*, 2012, pp. 796–799.

[9] ——, "Probabilistic sampling of finite renewal processes," *Bernoulli*, vol. 17, no. 4, pp. 1285–1326, 11 2011.

[10] ——, "Estimation of flow distributions from sampled traffic," *ACM Trans. Model. Perform. Eval. Comput. Syst.*, vol. 1, no. 3, pp. 11:1–11:28, May 2016.

[11] C. Estan and G. Varghese, "New directions in traffic measurement and accounting: Focusing on the elephants, ignoring the mice," *ACM Trans. Comput. Syst.*, vol. 21, no. 3, pp. 270–313, Aug. 2003.

[12] S. Venkataraman, D. Song, P. Gibbons, and A. Blum, "New streaming algorithms for superspreader detection," in *Proc. NDSS*, 2005.

[13] S. Ganguly and G. Cormode, "On estimating frequency moments of data streams," in *Proc. APPROX and RANDOM*. Springer-Verlag, 2007, pp. 479–493.

[14] H. Zhao, A. Lall, M. Ogihara, O. Spatscheck, J. Wang, and J. Xu, "A data streaming algorithm for estimating entropies of OD flows," in *Proc. ACM IMC*, 2007, pp. 279–290.

[15] A. Kumar, M. Sung, J. J. Xu, and J. Wang, "Data streaming algorithms for efficient and accurate estimation of flow size distribution," *SIGMETRICS Perform. Eval. Rev.*, vol. 32, no. 1, pp. 177–188, Jun. 2004.

[16] V. Karamcheti, D. Geiger, Z. Kedem, and S. Muthukrishnan, "Detecting malicious network traffic using inverse distributions of packet contents," in *Proc. ACM SIGCOMM MineNet*, 2005, pp. 165–170.

[17] A. Chen, Y. Jin, J. Cao, and L. Li, "Tracking long duration flows in network traffic," in *Proc. IEEE INFOCOM*, 2010, pp. 206–210.

[18] B. Whitehead, C.-H. Lung, and P. Rabinovitch, "An efficient hybrid approach to per-flow state tracking for high-speed networks," *Comput. Commun.*, vol. 36, no. 8, pp. 927 – 938, May 2013.

[19] P. Tune and D. Veitch, "Fisher information in flow size distribution estimation," *IEEE Trans. Info. Theory*, vol. 57, no. 10, pp. 7011–7035, Oct. 2011.

[20] N. Duffield, C. Lund, and M. Thorup, "Estimating flow distributions from sampled flow statistics," *IEEE/ACM Trans. Netw.*, vol. 13, pp. 933–946, Oct. 2005.

[21] N. B. Azzouna, F. Guillemin, S. Poisson, P. Robert, C. Fricker, and N. Antunes, "Inverting sampled ADSL traffic," in *Proc. IEEE ICC*, vol. 1, 2005, pp. 1–5.

[22] P. Hall, *An Introduction to the Theory of Coverage Processes*. Wiley, New York, 1998.

[23] N. H. Bingham and S. M. Pitts, "Non-parametric estimation for the $M/G/\infty$ queue," *Ann. Inst. Stat. Math.*, vol. 51, no. 1, pp. 71–97, 1999.

[24] P. Hall and J. Park, "Nonparametric inference about service time distribution from indirect measurements," *J. R. Stat. Soc. Series B*, vol. 66, no. 4, pp. 861–875, 2004.

[25] M. Brown, "An $M/G/\infty$ estimation problem," *Ann. Math. Statist.*, vol. 41, no. 2, pp. 651–654, 04 1970.

[26] U. N. Bhat, G. K. Miller, and S. S. Rao, "Statistical analysis of queuing system," in *Frontiers in queuing models and applications in Science and Engineering*, J. H. Dshalalow, Ed. CRC Press, 1997, ch. 13, pp. 351–394.