

Temporal Update Dynamics Under Blind Sampling

Xiaoyong Li, *Student Member, IEEE*, Daren B. H. Cline, and Dmitri Loguinov, *Senior Member, IEEE*

Abstract—Network applications commonly maintain local copies of remote data sources in order to provide caching, indexing, and data-mining services to their clients. Modeling performance of these systems and predicting future updates usually requires knowledge of the inter-update distribution at the source, which can only be estimated through blind sampling—periodic downloads and comparison against previous copies. In this paper, we first introduce a stochastic modeling framework for this problem, where updates and sampling follow independent point processes. We then show that all previous approaches are biased unless the observation rate tends to infinity or the update process is Poisson. To overcome these issues, we propose four new algorithms that achieve various levels of consistency, which depend on the amount of temporal information revealed by the source and capabilities of the download process.

Index Terms—Internet, network servers, storage area networks, web services, stochastic processes.

I. INTRODUCTION

MANY distributed systems in the current Internet manipulate objects that experience periodic modification in response to user actions, real-time events, data-centric computation, or some combination thereof. In these cases, each source (e.g., a webpage, DNS record, stock price) can be viewed as a stochastic process N_U that undergoes updates (i.e., certain tangible changes) after random delays U_1, U_2, \dots , which we assume have some empirical CDF $F_U(x)$. Note that $N_U(t)$ is a point process that counts the number of events in $[0, t]$; however, when t is not explicitly needed, we use N_U to represent the entire process.

Consistent estimation of inter-update distribution $F_U(x)$ is an important problem, whose solution yields not only better caching, replication [25], and allocation of download budgets [24], but also more accurate modeling and characterization of complex Internet systems [8], [10], [12]–[14], [23], [29], [31]–[33], [38], [40], [46], [48]. Similar issues arise in lifetime measurement, where U_i represents the duration of online presence for object or user i [6], [37], [41], [44].

The first challenge with measuring update-interval dynamics is to infer their distribution using *blind* sampling, where variables U_1, U_2, \dots are hidden from the observer. This scenario arises when the source can only be queried over the network using a point process N_S whose inter-download delays S_1, S_2, \dots have some empirical distribution $F_S(x)$. Due to bandwidth and/or CPU restrictions, a common requirement

is to bound $E[S_i]$ from below by some positive constant, which prevents infinitely fast sampling and introduces bias in the collected measurements. Unlike censored observations in statistics, which have access to truncated values of *each* U_i , the sampling process here has a tendency to miss entire update cycles and land in larger-than-average intervals, which gives rise to the inspection paradox [47].

The second challenge in blind sampling is to reconstruct the distribution of U_i from severely limited amounts of information available from each download. Specifically, the observer can only compare the two most-recent copies of the source and obtain indicator variables Q_{ij} of a change occurring between downloads i and j , for all $i < j$. This constraint is necessary because determining object-modification timestamps is a complicated endeavor. For example, dynamic webpages served by scripts are considered new on each download. Doing otherwise would require the server to store two copies of each page and compare them on the fly. Even with this approach, the exact modification time between two subsequent downloads remains unknown (e.g., a script rendering traffic maps would not know when congestion first occurred). This is further compounded by the fact that object updates are highly application-specific (e.g., search engines may remove ad banners, javascript, and other superfluous information before indexing).

Even if each observer's content-comparison algorithm could be uploaded to every source, the computational load needed to detect updates would make this service prohibitively expensive. This is especially true given the variety of pages and crawlers in the public Internet, each with its own variation of the algorithm. Besides complexity at the server, other issues include unwillingness of certain observers (e.g., commercial search engines) to disclose proprietary algorithms and difficulty of keeping them up-to-date on remote websites. As a result, variables $\{U_i\}$ are *hidden not just from the observer, but also the source*.

Existing studies on this topic [10], [19], [20], [26] use Poisson N_U and constant S_i . Due to the memoryless assumption on $F_U(x)$, the problem reduces to estimating just rate $\mu = 1/E[U_i]$, rather than an entire distribution, and many complex interactions between N_S and N_U are avoided in the analysis. However, more interesting cases arise in practice, where non-Poisson updates are quite common [4], [11], [21], [27]. Furthermore, guaranteeing constant S_i is impossible in certain applications where the return delay to the same object is computed in real-time and is governed by the properties of trillions of other sources (e.g., in search engines). Thus, new analytical techniques are required to handle such cases.

A. Contributions

Our first contribution is to formalize blind update sampling using a framework in which both N_U and N_S are general point

Manuscript received August 31, 2015; revised April 26, 2016; accepted June 4, 2016; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor S. P. Weber. Date of publication June 30, 2016; date of current version February 14, 2017. This work was supported by the National Science Foundation under Grants CNS-1017766 and CNS-1319984.

X. Li and D. Loguinov are with the Department of Computer Science and Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: xiaoyong@cse.tamu.edu; dmitri@cse.tamu.edu).

D. B. H. Cline is with the Department of Statistics, Texas A&M University, College Station, TX 77843 USA (e-mail: dcline@stat.tamu.edu).

Digital Object Identifier 10.1109/TNET.2016.2577680

processes. We establish performance objectives and specify assumptions necessary for this problem to be solvable. Our second contribution is to consider a simplified problem where the source provides last-modification timestamps for each download. This allows us to develop the necessary tools for tackling the more interesting cases that follow, build general intuition, consider conditions under which provably consistent estimation is possible, and explain the pitfalls of existing methods under non-Poisson updates.

Armed with these results, we next relax the availability of last-modified timestamps at the source. For situations where constant S_i is acceptable, we show that unbiased estimators developed earlier in the paper can be easily adapted to this environment and then suggest avenues for removing bias in traces collected by previous methods, all of which forms our third contribution. We finish the modeling part of the paper by considering random S_i and arrive at our fourth contribution, which is a novel method that can accurately reconstruct the update distribution under arbitrary $F_U(x)$ and mildly constrained $F_S(x)$.

Our last contribution is to apply our methods to Wikipedia logs, which contain all edit timestamps and allow easy construction of ground-truth $F_U(x)$. We show that the Poisson update assumption fails on the majority of the pages, with the remaining content having exponential inter-update delays but non-negligible correlation. Then, we compare the accuracy of our methods using the top 10 most-modified articles and develop a recommendation algorithm for selecting the best method for every possible scenario.

II. RELATED WORK

Analytical studies on estimating the update distribution under blind sampling have all assumed N_U was Poisson and focused on determining its average rate, i.e., μ for stationary cases [4], [10], [19], [20], [26] and $\mu(t)$ for non-stationary [39]. Extension to general processes was achieved by [26] under the assumption that sampling intervals S_i were infinitely small; however, the problem in these scenarios is trivial since every U_i is available to the observer with perfect accuracy.

In measurement literature, the majority of effort was spent on the behavior of web pages, including analysis of server logs [30], page-modification frequency during crawling [4], [7], [21], [27], RSS feed dynamics [38], and content change between consecutive observations [1], [18], [29]. Problems related to estimation of $F_U(x)$ have also emerged in prediction of future updates [8], [9], [16], [22], [35], [42], with a good survey in [28], and user lifetime measurement in decentralized P2P networks [6], [37], [41], [44].

III. OVERVIEW

This section introduces notation, formulates objectives, and lays down a roadmap of the studied methods.

A. Notation and Assumptions

Denote by u_i the time of the i -th update at the source. Define $N_U(t) = \max\{i : u_i \leq t\}$ to be the number of updates in the time interval $[0, t]$ and suppose $U_i = u_{i+1} - u_i$ is the

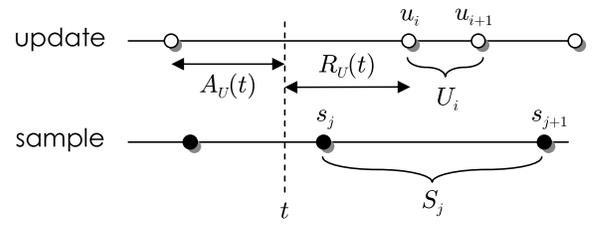


Fig. 1. Update/sample process notation.

i -th inter-update delay. Similarly, define s_j to be the j -th sampling point, $S_j = s_{j+1} - s_j$ to be the corresponding inter-sample delay, and $N_S(t) = \max\{j : s_j \leq t\}$ to be the number of samples in $[0, t]$. At time t , let age

$$A_U(t) = t - u_{N_U(t)} \quad (1)$$

and residual

$$R_U(t) = u_{N_U(t)+1} - t \quad (2)$$

be the backward/forward delays to the nearest update. These concepts are illustrated in Fig. 1. Note that interval U_i in the figure cannot be seen or measured by the observer, which is why we called it “hidden” earlier.

We adopt the sample-path approach of [24] to model both processes, which needs the following assumption.

Assumption 1: Both N_U and N_D are age-measurable.

While quite technical [24], this condition means that the age of each process examined at a random time $t \in [0, T]$ is well-defined as $T \rightarrow \infty$. Age-measurability generalizes well-known families of processes (e.g., renewal, regenerative, ergodic) and is the weakest set of conditions under which N_U can be meaningfully sampled to produce a distribution of its cycle lengths. Defining $\mathbf{1}_A$ to be the indicator variable of event A , Assumption 1 guarantees existence of [24]:

$$F_U(x) := \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{U_i \leq x} \quad (3)$$

and

$$F_S(x) := \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{j=1}^n \mathbf{1}_{S_j \leq x}, \quad (4)$$

which are empirical distributions of inter-update and inter-sample delays, respectively. This allows us to use random variables $U \sim F_U(x)$ and $S \sim F_S(x)$ to represent update/sample cycle durations, where $\mu = 1/E[U]$ and $\lambda = 1/E[S]$ are the corresponding rates.

Suppose A_U and R_U are the equilibrium versions of $A_U(t)$ and $R_U(t)$, respectively, as $t \rightarrow \infty$. From [24], they have the well-known residual CDF:

$$G_U(x) := \mu \int_0^x (1 - F_U(y)) dy, \quad (5)$$

whose density is $g_U(x) := G'_U(x) = \mu(1 - F_U(x))$. We set the goal of the sampling process to determine the distribution $F_U(x)$ based on observations at times $\{s_j\}_{j \geq 1}$, i.e., using a single realization of the system.

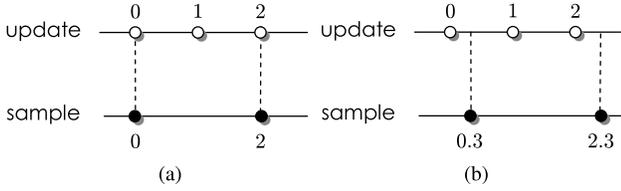


Fig. 2. Example of phase-lock that prevents recovery of $F_U(x)$. (a) $s_1 = 0$. (b) $s_1 = 0.3$.

B. Applications

Knowledge of $F_U(x)$ enables performance analysis in many fields that employ lazy (i.e., pull-based) data replication. For example, search engines implement a sampling process N_S using crawlers that periodically revisit web content and merge updates into backend databases. These organizations are often concerned with staleness of pages in their index and the probability that users encounter outdated results. In order to determine the download frequency needed to maintain staleness below a certain threshold, the expected number of updates by which the index is trailing the source, or the amount of bandwidth needed to maintain a collection of pages at some freshness level, accurate knowledge of source dynamics, i.e., distribution $F_U(x)$, is required [24].

In another example, suppose a data center replicates a quickly changing database (driven by some update process N_U) among multiple nodes for scalability and fault-tolerance reasons. Because of the highly dynamic nature of the source, individual replicas may not stay fresh for long periods of time, but they jointly may offer much better performance as a whole. In such cases, questions arise about the number of replicas k that should be queried by clients to obtain results consistent with the source [25] and/or the probability that a cluster of n replicas can recover the most-recent copy of the source when it crashes [24]. Similar problems appear in multi-hop replication and cooperative caching, where service capacity of the caching network is studied as well [25].

Finally, accurate measurement of $F_U(x)$ enables better characterization of Internet systems, their update patterns in response to external traffic, and even user behavior. While it is possible to use the exponential distribution to approximate any $F_U(x)$, as typically done in the literature [4], [10], [19], [20], [26], this can lead to significant errors in the analysis. As shown in [24] using the search-engine example and Wikipedia’s update process N_U , the exponential assumption may produce errors in the download bandwidth that are two orders of magnitude. In more complicated settings, such as cascaded and cooperative systems [25], the impact of inaccurate $F_U(x)$ may be even higher.

C. Caveats

In general, sample-path approaches to modeling interaction of two processes lead to a possibility of *phase-lock*, where the distance of download points from the last update, i.e., $\{A_U(s_j)\}_{j \geq 1}$, is not a mixing process. For example, consider Fig. 2, where $U_i = 1$ for $i \geq 1$ and $S_j = 2$ for $j \geq 1$. Notice that update ages $A_U(s_j)$ observed at download points are all equal to s_1 . Since this case cannot be distinguished from $U_i = 0.5$ or $U_i = 2$, it is easy to see how phase-lock

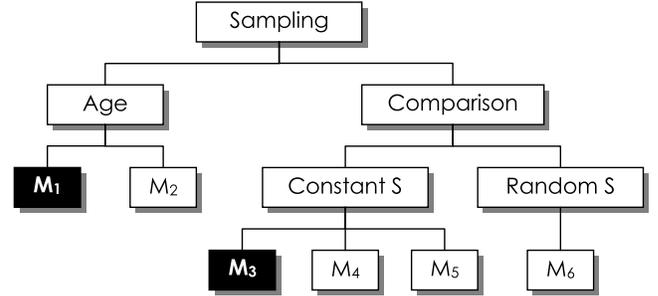


Fig. 3. Method taxonomy (shaded boxes indicate Poisson-only techniques).

precludes estimation of $F_U(x)$. The problem can be avoided by requiring that the considered cycle lengths exhibit certain mixing properties.

Definition 1: A random variable X is called *lattice* if there exists a constant c such that X/c is always an integer, i.e., $\sum_{i=1}^{\infty} P(X/c = i) = 1$.

Lattice distributions are undesirable in our context as they produce phase-lock when sampling other lattice distributions. Finding the most general conditions for avoidance of phase-lock is a difficult problem, but the following sufficient condition exists [24].

Definition 2: An age-measurable point process is called *age-mixing* if it is renewal with non-lattice delays.

Assumption 2: At least one of (N_U, N_S) is age-mixing.

This condition is easy to satisfy with any continuous random variable, but a more esoteric example would be a discrete distribution placing mass on two numbers whose ratio is irrational, e.g., $(\pi, 3)$ or $(e, \sqrt{2})$.

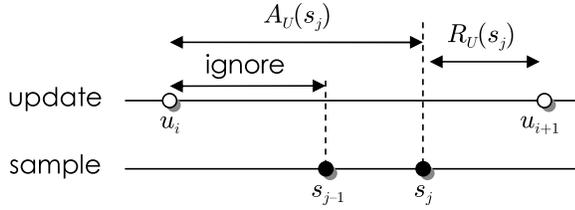
D. Roadmap

As illustrated in Fig. 3, we partition the various approaches into two broad categories. In *age* sampling, the observer has access to the last-modified timestamp $u_{N_U}(s_j)$ at each download point s_j , or equivalently, the update age $A_U(s_j)$. Although now rare, this information can still be sometimes obtained from the HTTP headers, timestamps within the downloaded HTML, or sitemaps [28]. As shown in the figure, we call the two studied methods in this category M_1 and M_2 . They operate by deriving $F_U(x)$ from the collected age samples, where M_1 has been proposed in previous work [10], [26] for Poisson-only cases and M_2 is novel.

In *comparison* sampling, we assume that the observer retains the most recent copy of the object or a fingerprint of its relevant portions (e.g., after removing ads and repeated keywords). Define Q_{ij} to be an update-indicator process:

$$Q_{ij} = \begin{cases} 1 & \text{update occurs between } s_i \text{ and } s_j \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Unlike the previous scenario, estimation of $F_U(x)$ here must use only binary values $\{Q_{ij}\}$. Going back to Fig. 3, we study comparison sampling under two strategies. For *constant S*, we first analyze two methods we call M_3 and M_4 , which are discrete versions of M_1 and M_2 , respectively. We then propose a novel method M_5 that allows recovery of $F_U(x)$ from biased samples of M_3 , should such traces become available from prior measurement studies. For *random S*, we introduce our

Fig. 4. Illustration of M_1 .

final approach M_6 that is consistent under the most general conditions.

IV. AGE SAMPLING

This section is a prerequisite for the results that follow. It starts with understanding state of the art in this field and its pitfalls. It then shows that a simple modification allows prior work to become unbiased under non-Poisson updates.

A. Basics

In age sampling, the observer has a rich amount of information about the update cycles. This allows reconstruction of $F_U(x)$ in all points $x \geq 0$, which we set as our goal.

Definition 3: Suppose $H(x, T)$ is a CDF estimator that uses observations in $[0, T]$. Then, we call it *consistent* with respect to distribution $F(x)$ if it converges in probability to $F(x)$ as the sampling window becomes large:

$$\lim_{T \rightarrow \infty} H(x, T) = F(x), \quad x \geq 0. \quad (7)$$

Note that consistent estimation of $F_U(x)$ is equivalent to that of $G_U(x)$ since there is a one-to-one mapping (5) between the two functions. Specifically, appropriately smoothing the numerical derivative of a consistent estimate for $G_U(x)$ can provide a consistent estimate for $g_U(x) = G'_U(x)$, from which $F_U(x) = 1 - g_U(x)/g_U(0)$ follows. Furthermore, the update rate μ is also readily available as $g_U(0)$. The rate of convergence for the two CDFs may be different, but asymptotically this makes no difference. Under Poisson N_U , the memoryless property ensures that $F_U(x) = G_U(x)$ and no conversion is needed; however, in more general cases, this distinction is important.

B. Modeling M_1

To estimate the mean μ of a Poisson update process, prior studies [10], [26] proposed that only a subset of age samples $\{A_U(s_j)\}_{j \geq 1}$ be retained by the observer. In particular, when multiple sample points land in the same update interval, only the one with the largest age is kept, while the others are discarded. As shown in Fig. 4, points s_{j-1} and s_j hit the same update cycle $[u_i, u_{i+1}]$, in which case only $A_U(s_j)$ is used in the measurement and $A_U(s_{j-1})$ is ignored. It was perceived in [10] and [26] that doing otherwise would create a bias and lead to incorrect estimation, but no proof was offered. We call this method M_1 and study its performance next.

Although previous studies [10], [26] mainly focused on M_1 under constant S , we consider it as a general random variable. As we prove in this section, M_1 is a fascinating method because depending on S it can measure an entire spectrum of distributions contained between $F_U(x)$ and $G_U(x)$. From

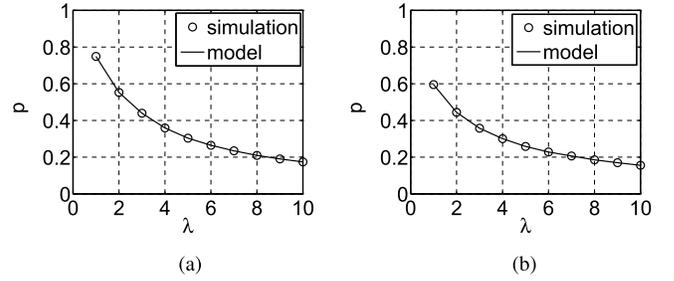
Fig. 5. Verification of (9) under Pareto U ($\mu = 2$). (a) constant S . (b) exponential S .

Fig. 4, notice that M_1 collects ages $A_U(s_j)$ at such points s_j that satisfy $R_U(s_j) < S_j$, or equivalently $Q_{j,j+1} = 1$. All other age measurements are ignored. Then, the fraction of age samples retained by M_1 in $[0, T]$ is given by:

$$p_T := \frac{1}{N_S(T)} \sum_{j=1}^{N_S(T)} \mathbf{1}_{R_U(s_j) < S_j}, \quad (8)$$

which is an important metric that determines the overhead of M_1 and its bias later in the section. Expansion of (8) in the next result follows from Assumption 2, the equilibrium residual equation for non-lattice intervals, and the law of large numbers [47].

Theorem 1: As $T \rightarrow \infty$, p_T converges in probability to:

$$p := \lim_{T \rightarrow \infty} p_T = P(R_U < S) = E[G_U(S)]. \quad (9)$$

This result shows that p is affected not just by the update distribution $F_U(x)$, but also the sample distribution $F_S(x)$. To see this effect in simulations, we use constant and exponential S to sample Pareto $F_U(x) = 1 - (1+x/\beta)^{-\alpha}$, where $\alpha = 3$ and $\beta = 1$ throughout the paper. Fig. 5 confirms a good match between the model and simulations. As expected, p decreases as the sampling rate $\lambda = 1/E[S]$ increases, which is caused by an increased density of points landing within each update interval and thus a higher discard rate. The figure also shows that constant S samples more points than the exponential case. In fact, it is possible to prove a more general result – constant S exhibits the largest p (i.e., highest overhead) for a given λ – but this is not essential to our results below.

Let $K(x, T)$ be the number of samples that M_1 obtains in $[0, T]$ with values no larger than x :

$$K(x, T) := \sum_{j=1}^{N_S(T)} \mathbf{1}_{R_U(s_j) < S_j} \mathbf{1}_{A_U(s_j) \leq x}. \quad (10)$$

Then, it produces a distribution in $[0, T]$ given by:

$$G_1(x, T) := \frac{K(x, T)}{K(\infty, T)}. \quad (11)$$

Theorem 2: Denoting by $\bar{F}(x) = 1 - F(x)$ the complement of function $F(x)$ and letting $T \rightarrow \infty$, the tail distribution of the samples collected by M_1 converges in probability to:

$$\bar{G}_1(x) := \lim_{T \rightarrow \infty} \bar{G}_1(x, T) = \frac{E[G_U(x+S) - G_U(x)]}{E[G_U(S)]}. \quad (12)$$

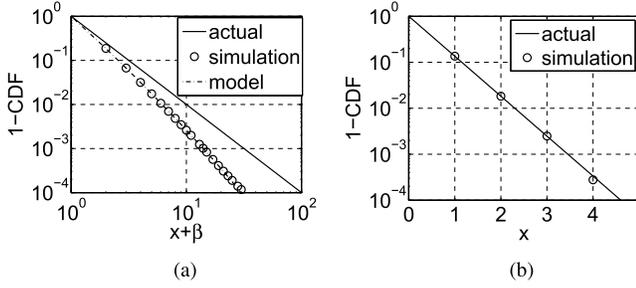


Fig. 6. Simulation results of M_1 under exponential S ($\lambda = 1, \mu = 2$). (a) Pareto U . (b) exponential U .

Proof: Under Assumption 2 and $T \rightarrow \infty$, $A_U(s_j)$ and $R_U(s_j)$ converge to their equilibrium versions A_U and R_U , respectively. Therefore:

$$\lim_{T \rightarrow \infty} \frac{K(x, T)}{N_S(T)} = P(A_U \leq x, R_U < S). \quad (13)$$

From Theorem 1, we know that:

$$\lim_{T \rightarrow \infty} \frac{K(\infty, T)}{N_S(T)} = p = E[G_U(S)]. \quad (14)$$

Dividing (13) by (14) yields:

$$G_1(x) = \lim_{T \rightarrow \infty} G_1(x, T) = \frac{P(A_U \leq x, R_U < S)}{E[G_U(S)]}, \quad (15)$$

where $E[G_U(S)] > 0$ is guaranteed for all cases except S being zero with probability 1. To derive the numerator of (15), condition on R_U and S :

$$\begin{aligned} P(A_U \leq x, R_U < S) \\ = \int_0^\infty \left[\int_0^z P(A_U \leq x | R_U = y) g_U(y) dy \right] dF_S(z), \end{aligned} \quad (16)$$

Expanding the probability of event $A_U \leq x$ given a fixed residual $R_U = y$ leads to:

$$\begin{aligned} P(A_U \leq x | R_U = y) &= \frac{P(y < U \leq x + y)}{P(U > y)} \\ &= \frac{F_U(x + y) - F_U(y)}{1 - F_U(y)}. \end{aligned} \quad (17)$$

Recalling that $g_U(y) = \mu(1 - F_U(y))$ is the residual density and applying (17), the inside integral of (16) becomes:

$$\begin{aligned} \int_0^z P(A_U \leq x | R_U = y) g_U(y) dy \\ = \mu \int_0^z (F_U(x + y) - F_U(y)) dy \\ = \mu \int_0^z \bar{F}_U(y) dy - \mu \int_x^{z+x} \bar{F}_U(w) dw \\ = G_U(z) + G_U(x) - G_U(x + z). \end{aligned} \quad (18)$$

This transforms (15) to:

$$\begin{aligned} G_1(x) &= \frac{\int_0^\infty (G_U(z) + G_U(x) - G_U(x + z)) dF_S(z)}{E[G_U(S)]} \\ &= \frac{E[G_U(S) - G_U(x + S)] + G_U(x)}{E[G_U(S)]}, \end{aligned} \quad (19)$$

which is the complement of the tail in (12). \blacksquare

Observe from (12) that M_1 measures neither the update distribution $F_U(x)$ nor the age distribution $G_U(x)$. To see the extent of this bias, Fig. 6(a) plots simulation results for exponential S and Pareto U in comparison to (12). Notice in the figure that our model closely tracks the simulated tail $\bar{G}_1(x)$, which remains heavy-tailed, albeit different from that of the target distribution $F_U(x)$. It was known in prior work [10], [26] that M_1 is unbiased for exponential U , with one example illustrated in Fig. 6(b). We next strengthen this result to provide not only a sufficient, but also a necessary condition.

Theorem 3: Exponential U is the only case that allows M_1 to be consistent with respect to $F_U(x)$ for all S .

Proof: For M_1 to be consistent in (12), $E[G_U(x + S) - G_U(x)] = E[G_U(S)]\bar{F}_U(x)$ must hold for all distributions S . This is a well-known functional equation that is solved only by exponential $F_U(x)$. \blacksquare

C. Quantifying Bias in M_1

Suppose $D_1 \sim G_1(x)$ is a random variable drawn from the distribution observed by M_1 over an infinitely long measurement period. Our goal in this subsection is to determine the relationship between D_1 , U , and A_U under different sampling rates and update distributions. We first re-write (12) in a more convenient form.

Theorem 4: The tail distribution measured by M_1 can be expressed in two alternative forms:

$$\bar{G}_1(x) = \bar{G}_U(x) \frac{P(A_U < x + S | A_U > x)}{P(A_U < S)} \quad (20)$$

$$= \bar{F}_U(x) \frac{E[\int_0^S P(U > x + y | U > x) dy]}{E[\int_0^S P(U > y) dy]}. \quad (21)$$

Proof: We first show (20). Recalling that $G_U(x) = P(A_U < x)$ yields:

$$\begin{aligned} \bar{G}_1(x) &= \frac{P(A_U < x + S) - P(A_U < x)}{P(A_U < S)} \\ &= \bar{G}_U(x) \frac{P(x < A_U < x + S)}{P(A_U < S)P(A_U > x)}. \end{aligned} \quad (22)$$

From the definition of conditional probability, we get:

$$\frac{P(x < A_U < x + S)}{P(A_U > x)} = P(A_U < x + S | A_U > x). \quad (23)$$

Substituting (23) into (22), we get (20).

To establish (21), rewrite (22) as:

$$\bar{G}_1(x) = \bar{F}_U(x) \frac{P(x < A_U < x + S)}{P(A_U < S)P(U > x)}, \quad (24)$$

whose numerator can be transformed to:

$$\begin{aligned} P(x < A_U < x + S) &= \mu E \left[\int_x^{x+S} \bar{F}_U(y) dy \right] \\ &= \mu E \left[\int_0^S \bar{F}_U(x + y) dy \right], \end{aligned} \quad (25)$$

where we use the fact that $g_U(x) = \mu\bar{F}_U(x)$. Dividing (25) by $\bar{F}_U(x)$ produces:

$$\frac{P(x < A_U < x + S)}{P(U > x)} = \mu E \left[\int_0^S P(U > x + y | U > x) dy \right].$$

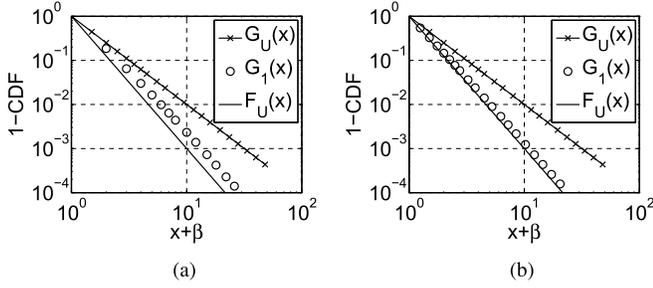


Fig. 7. Tail sandwich of M_1 under Pareto updates and constant S ($\mu = 2$). (a) $\lambda = 1$. (b) $\lambda = 4$.

Similarly, we can expand:

$$P(A_U < S) = \mu E \left[\int_0^S P(U > x) dy \right]. \quad (26)$$

Using the last two equations in (24), we obtain (21). ■

Theorem 4 suggests that the tail of D_1 may indeed have some relationship to those of A_U and U . In order to establish this formally, we need to define three classes of variables.

Definition 4: Variable X is said to be NWU (new worse than used) if $P(X > x + y | X > y) \geq P(X > x)$ for all $x, y \geq 0$. If this inequality is reversed, X is said to be NBU (new better than used). Finally, if $P(X > x + y | X > y) = P(X > x)$ for all $x, y \geq 0$, the variable is called *memoryless*.

Note that NWU distributions are usually heavy-tailed, with two common representatives being Pareto and Weibull. Conditioning on U 's survival to some age y , its residual length $U - y$ is stochastically larger than U itself. NBU are typically light-tailed distributions, exemplified by uniform and constant. Finally, the memoryless class consists of only exponential distributions, where past knowledge has no effect on the future.

When both U and A_U are NWU, as is the case with Pareto distributions, Theorem 4 shows that $\bar{G}_1(x)$ is “sandwiched” between the other two tails, i.e., $\bar{F}_U(x)$ serves as a lower bound and $\bar{G}_U(x)$ as an upper. This means that D_1 is stochastically smaller than A_U , but stochastically larger than U . Fig. 7 shows an example confirming this, where the faster sampling rate in (b) moves the curve closer to $\bar{F}_U(x)$. The relationship among the tails is reversed if U and A_U are NBU. For exponential update distributions, all three tails are equal, which is another way to show its lack of bias. We examine a few other cases next.

D. Achieving Consistency in M_1

If $\bar{G}_1(x)$ is sandwiched between two tails, the first intuitive avenue for removing bias is to tighten the distance between $\bar{F}_U(x)$ and $\bar{G}_U(x)$; however, this can only be achieved by forcing the source to undergo updates with U that is “closer” to exponential. As this is usually impractical, the second technique is to adjust the sampling distribution $F_S(x)$ such that the distance of $\bar{G}_1(x)$ to one of U 's tails shrinks to zero. To this end, our next result demonstrates that D_1 “leans” towards U or A_U solely based on the fraction of retained samples p .

Theorem 5: For $p \rightarrow 1$, variable D_1 sampled by M_1 converges in distribution to A_U . For $p \rightarrow 0$ and mild conditions on S , variable D_1 converges in distribution to U .

Proof: Recall that $p = E[G_U(S)]$. When $E[G_U(S)] \rightarrow 1$, so does $E[G_U(S + x)]$. Therefore:

$$\bar{G}_1(x) = \frac{E[G_U(S + x)] - G_U(x)}{E[G_U(S)]} \rightarrow \bar{G}_U(x). \quad (27)$$

To prove the second part, assume that $S/E[S]$ converges to a random variable with mean 1. Since $p \rightarrow 0$ implies that $S \rightarrow 0$ almost surely, we get:

$$\frac{G_U(S)}{E[S]} = \frac{\int_0^S \bar{F}_U(y) dy}{E[U]E[S]} = \frac{S \int_0^1 \bar{F}_U(Sy) dy}{E[U]E[S]} \rightarrow \mu, \quad (28)$$

where we use the fact that $\bar{F}_U(Sy) \rightarrow 1$ for all fixed y .

Noticing that $G_U(S)/E[S]$ is upper bounded by random variable $\mu S/E[S]$, the latter of which has a finite mean, and applying the dominated convergence theorem (DCT), we get:

$$\lim_{p \rightarrow 0} \frac{E[G_U(S)]}{E[S]} = \mu. \quad (29)$$

Similarly, we obtain:

$$\begin{aligned} \frac{G_U(S + x) - G_U(x)}{E[S]} &= \frac{\int_x^{S+x} \bar{F}_U(y) dy}{E[U]E[S]} \\ &= \frac{S \int_0^1 \bar{F}_U(Sy + x) dy}{E[U]E[S]}, \end{aligned} \quad (30)$$

which converges to $\mu \bar{F}_U(x)$. Applying the DCT again, we get:

$$\lim_{p \rightarrow 0} \frac{E[G_U(S + x) - G_U(x)]}{E[S]} = \mu \bar{F}_U(x). \quad (31)$$

Combining (29) and (31) produces:

$$\lim_{p \rightarrow 0} \frac{E[G_U(S + x) - G_U(x)]}{E[G_U(S)]} = \bar{F}_U(x), \quad (32)$$

which is what we intended to prove. ■

To understand this result, we discuss several examples. In order to converge p to 1, method M_1 has to sample with sufficiently large S to achieve $P(S > R_U) = 1$. For general $F_U(x)$, this can be guaranteed only if S converges to infinity, in which case the measurement process will be impossibly slow. If an upper bound on U is known, then setting S to be always larger can also produce $p = 1$. In these scenarios, however, M_1 will sample $G_U(x)$ and additional steps to recover $F_U(x)$ must be undertaken.

To achieve $p = 0$, M_1 has to use high sampling rates such that each update interval contains an infinite number of samples, i.e., S must converge to zero. In this case, the method may consume exorbitant network resources and additionally create undesirable load conditions at the source.

E. Method M_2

Instead of using the largest age sample for each detected update, a more sound option is to use *all* available ages. While extremely simple, this method has not been proposed before. We call this strategy M_2 and define $G_2(x, T)$ to be the fraction of its samples in $[0, T]$ with values smaller than or equal to x :

$$G_2(x, T) := \frac{1}{N_S(T)} \sum_{j=1}^{N_S(T)} \mathbf{1}_{A_U(s_j) \leq x}. \quad (33)$$

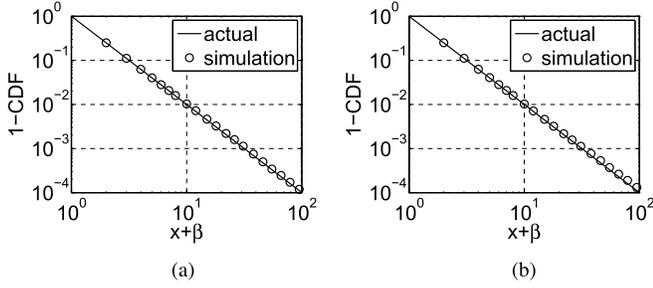


Fig. 8. Verification of (34) under Pareto updates and $\lambda = 1$. (a) exponential S . (b) constant S .

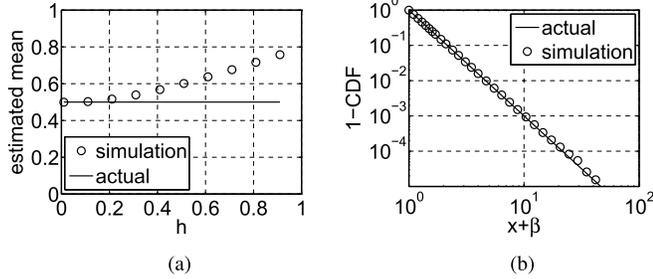


Fig. 9. Performance of M_2 under Pareto U and constant S ($\mu = 2, \lambda = 100$). (a) $E[U]$ (b) $F_U(x)$ ($h = 0.1$).

The next result follows from Assumption 2 and the equilibrium residual equation [24].

Theorem 6: Method M_2 is consistent with respect to the age distribution:

$$G_2(x) := \lim_{T \rightarrow \infty} G_2(x, T) = G_U(x). \quad (34)$$

Next we use simulations to verify the usefulness of (34). From Fig. 8, observe that the sampled distribution of M_2 does in fact equal $G_U(x)$. To obtain $F_U(x) = 1 - g_U(x)/g_U(0)$ from an empirical CDF $G_U(x)$, we adopt numerical differentiation from [43]. This method uses bins of size h and k -point derivatives, bounding Taylor-expansion errors to $O(h^k/k!)$. For the estimator to work, it must first accurately determine $g_U(0) = 1/E[U]$. Using $k = 5$ and non-symmetric (i.e., one-sided) derivatives around $x = 0$, Fig. 9(a) demonstrates that the estimated $E[U]$ monotonically decreases in h and eventually stabilizes at the true value. Since h is a user-defined parameter independent of (N_U, N_S) , it can be arbitrarily small. Thus, a binary search on h to find the flat region in $E[U]$ can always determine its value with high accuracy. Applying this technique, the update distribution estimated by M_2 is shown in Fig. 9(b) in comparison to $F_U(x)$. Notice that the two curves are indistinguishable.

F. Discussion

Although M_1 has fewer samples, its network traffic remains the same as that of M_2 , because they both have to contact the source $N_S(t)$ times in $[0, t]$. However, the smaller number of *retained* values in M_1 may lead to lower computational cost and better RAM usage in density-estimation techniques that utilize all available samples (e.g., kernel estimators). For the route we have taken, i.e., differentiation of $G_2(x)$, the two methods exhibit the same overhead.

We now focus on the performance of M_2 in finite observation windows $[0, T]$. One potential issue is the redundancy

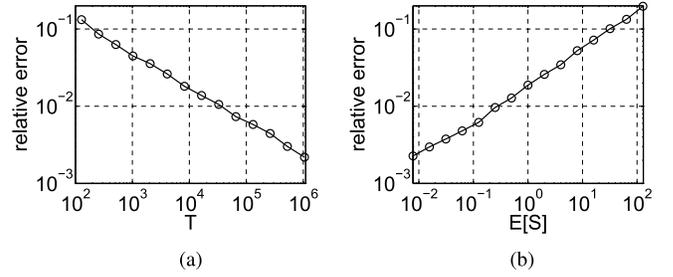


Fig. 10. Average relative error of $\zeta(T)$ of M_2 under Pareto U and exponential S ($\mu = 2, m = 1000$). (a) impact of T ($\lambda = 1$). (b) impact of S ($T = 10K$).

(and high dependency) of samples that it collects (i.e., all ages within the same update interval are deterministically predictable), which is what M_1 tried to avoid. While necessary, can this redundancy lead to slower convergence? For a given T , would it be better to collect fewer samples that are spaced further apart?

Define

$$\zeta(T) := \frac{1}{N_S(T)} \sum_{j=1}^{N_S(T)} A_U(s_j) \quad (35)$$

to be the average age observed by M_2 in $[0, T]$ using one realization of the system. We now use deviation of $\zeta(T)$ from $E[A_U] = \mu E[U^2]/2$ as indication of error. Specifically, let

$$\epsilon(T) := E \left[\left| 1 - \frac{\zeta(T)}{E[A_U]} \right| \right]. \quad (36)$$

be the expected relative error computed over m sample-paths.

First, we fix the sampling rate $\lambda = 1$ and change T from 100 to 1M time units. As expected, $\epsilon(T)$ in Fig. 10(a) monotonically decreases as the observation window gets larger, confirming asymptotic convergence of M_2 discussed throughout this section. Next, we keep T constant at 10K and vary $E[S]$. As shown in Fig. 10(b), the error monotonically drops with $E[S]$, suggesting that having more samples, regardless of how redundant, improves performance.

V. COMPARISON SAMPLING: CONSTANT INTERVALS

In contrast to the previous section, the remaining methods do not have access to age; instead, they must work with binary observations Q_{ij} , which indicate whether an update occurred between two sampling points s_i and s_j . This section deals with constant inter-download delay, which is not just simple to implement and the only one considered in the literature, but also maximally polite (i.e., least bursty) for a given download rate λ .

A. Basics

Assume constant inter-sample delays $S = \Delta$ and notice that all observations related to update intervals must be multiples of Δ . It is therefore impossible to reconstruct $F_U(x)$, or even $G_U(x)$, in every point x . This requires an adjustment in our objectives.

Definition 5: An estimator $H(x, T)$ is Δ -consistent with respect to distribution $F(x)$ if it can correctly reproduce it in all discrete points $x_n = n\Delta$ as $T \rightarrow \infty$:

$$\lim_{T \rightarrow \infty} H(x_n, T) = F(x_n), \quad n = 1, 2, \dots \quad (37)$$

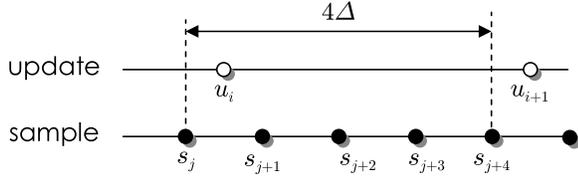


Fig. 11. Comparison sampling in M_3 with constant intervals of size Δ .

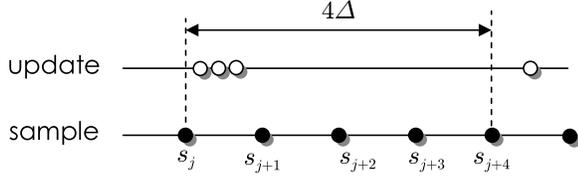


Fig. 12. Pitfalls of M_3 .

Unless the sampling rate is infinite, Δ -consistent estimation produces a step-function CDF. The main caveat of this section is that knowledge of the age distribution in discrete points is insufficient for Δ -consistent estimation of $F_U(x)$. This occurs because the estimated $G_U(x)$ lacks data in every interval (x_n, x_{n+1}) , which precludes differentiation and leaves $g_U(x)$ unobtainable.

Depending on the smoothness of $G_U(x)$ and/or prior knowledge about the target distribution, one can use interpolation between the known points $G_U(x_n)$. In such cases, $F_U(x)$ may be reconstructed with high accuracy using kernel density-estimation techniques; however, the result is application-specific. We thus do not dwell on numerical methods needed to perform these manipulations and instead focus on Δ -consistency in regard to $G_U(x)$.

B. Method M_3

Prior work in several fields [6], [10], [26], [37], [41] has suggested an estimator, which we call M_3 , that rounds the distance between each adjacent pair of detected updates to the nearest multiple of Δ , from which it builds a distribution $G_3(x)$. This technique was used in [10] and [26] to track webpage updates, in [36] to estimate lifetimes of storage objects, and in [6], [37], and [41] to sample user lifetimes in P2P networks. In the OS/networking literature, the approach is known as *Create-Based Method* (CBM) because it tracks each object from its creation, as opposed to other methods that track deletions.

Define r_k to be the number of samples needed to see the k -th update, i.e.,

$$r_k := \min \left\{ m \geq 1 : \sum_{j=1}^m Q_{j,j+1} = k \right\}. \quad (38)$$

Then, the samples collected by M_3 are $(r_{k+1} - r_k)\Delta$ for $k = 1, 2, \dots$. To understand this better, Fig. 11 shows an example where updates are detected after downloads j and $j+4$, which produces $r_{k+1} - r_k = 4$ and a single sample 4Δ . Based on the description in prior work, this technique serves the purpose of directly measuring U_i by counting full intervals

of size Δ that fit in $[u_i, u_{i+1}]$. As a result, the output of M_3 is usually expected to produce the update distribution $F_U(x)$.

While this makes sense for the case in Fig. 11, the method becomes grossly inaccurate when multiple updates occur within Δ time units of each other, which brings us back to the issue of hidden variables U_i . Consider Fig. 12, where $2/3$ of the update durations are less than Δ . Since M_3 in this scenario produces one sample 4Δ , it skews the mass of the distribution to much higher values than needed.

We now model the performance of M_3 under general U and obtain the limiting distribution of its samples. Define $G_3(x, T)$ to be the CDF of observed durations in $[0, T]$:

$$G_3(x, T) := \frac{\sum_{k=1}^{\infty} \mathbf{1}_{r_k \leq T} \mathbf{1}_{(r_{k+1} - r_k)\Delta \leq x}}{\sum_{k=1}^{\infty} \mathbf{1}_{r_k \leq T}}. \quad (39)$$

Let $x^+ = \Delta \lceil x/\Delta \rceil$ be x rounded-up to the nearest multiple of Δ and consider the following result.

Theorem 7: The tail distribution of M_3 is a step-function:

$$\bar{G}_3(x_n) := \lim_{T \rightarrow \infty} \bar{G}_3(x_n, T) = \frac{G_U(x_{n+1}) - G_U(x_n)}{G_U(\Delta)}. \quad (40)$$

Proof: Notice from Fig. 12 that age samples collected by M_3 can be viewed as discrete versions of those in M_1 . Indeed, the sample obtained by M_3 at any download instance s_j is $A_U^+(s_j)$. Since condition $A_U^+(s_j) < x_n$ is equivalent to $A_U(s_j) < x_n$ for $x_n = n\Delta$, we obtain:

$$G_3(x_n, T) = \frac{\sum_{j=1}^{N_S(T)} \mathbf{1}_{R_U(s_j) < s_j} \mathbf{1}_{A_U(s_j) \leq x_n}}{\sum_{j=1}^{N_S(T)} \mathbf{1}_{R_U(s_j) < s_j}}, \quad (41)$$

which is exactly the same as $G_1(x_n, T)$ in (11). Therefore, the tail of $G_3(x_n, T)$ converges to the result in (12), with S replaced by Δ . Doing so produces (40). Since $G_3(x)$ has no information between discrete points x_n , it must be constant in each interval $[x_n, x_{n+1})$, which means it is a step-function. ■

Define a random variable $D_3 \sim G_3(x)$. With the result above, its average becomes readily available.

Theorem 8: The expectation of D_3 is given by:

$$E[D_3] = \frac{\Delta}{G_U(\Delta)}. \quad (42)$$

Proof: It is well-known that the mean of a non-negative lattice random variable can be obtained by summing up its tail distribution:

$$E[D_3] = \Delta \sum_{n=0}^{\infty} \bar{G}_3(x_n). \quad (43)$$

Expanding $\bar{G}_3(x_n)$ using (40) and canceling all but two remaining terms leads to the desired result. ■

Given the discussion in the proof of Theorem 7, consistency and limitations of method M_3 are pretty similar to those of M_1 .

Corollary 1: Exponential is the only update distribution that allows M_3 to be Δ -consistent with respect to $F_U(x)$ for all Δ . Furthermore, Δ -consistency with $G_U(x)$ can be achieved using $\lambda \rightarrow 0$ and with $F_U(x)$ using $\lambda \rightarrow \infty$.

C. Method M_4

Using the rationale behind M_2 , we now propose another new method, which we call M_4 . At each sampling point s_j , the obtained value is:

$$a_j := \begin{cases} \Delta & Q_{j-1,j} = 1 \\ a_{j-1} + \Delta & \text{otherwise} \end{cases}. \quad (44)$$

For the example in Fig. 11, this method observes Δ in point s_{j+1} , 2Δ in s_{j+2} , 3Δ in s_{j+3} and 4Δ in s_{j+4} , resetting back to Δ in s_{j+5} . Denote by:

$$G_4(x, T) = \frac{1}{N_S(T)} \sum_{j=1}^{N_S(T)} \mathbf{1}_{a_j \leq x} \quad (45)$$

the distribution generated by M_4 in $[0, T]$. Then, we have the following result.

Theorem 9: Method M_4 is Δ -consistent with respect to the age distribution:

$$G_4(x_n) := \lim_{T \rightarrow \infty} G_4(x_n, T) = G_U(x_n). \quad (46)$$

Proof: It is not difficult to see that M_4 collects samples $A_U^+(s_j)$ in all points s_j , where $x^+ = \Delta \lceil x/\Delta \rceil$ as before. Therefore,

$$G_4(x_n, T) = \frac{\sum_{j=1}^{N_S(T)} \mathbf{1}_{A_U^+(s_j) \leq x_n}}{N_S(T)} \quad (47)$$

Since the CDF is computed only in discrete points x_n , the above can be written as:

$$G_4(x_n, T) = \frac{\sum_{j=1}^{N_S(T)} \mathbf{1}_{A_U(s_j) \leq x_n}}{N_S(T)} = G_2(x_n, T), \quad (48)$$

which converges to $G_U(x_n)$ using (34). ■

Define a random variable $D_4 \sim G_4(x)$, where $G_4(x)$ is a right-continuous step-function taking jumps at each x_n . Interestingly, even though M_3 keeps the largest age sample in each detected update interval $[u_i, u_{i+1}]$, the mean of its values $E[D_3]$ is not necessarily larger than that of D_4 . For example, with Pareto updates and $\Delta = 1$, we get $E[D_4] = 1.63$ and $E[D_3] = 1.33$. This can be explained by our previous discussion showing that under NWU update intervals the tail $\bar{G}_3(x)$ is upper-bounded by $\bar{G}_4(x)$, which implies $E[D_4] \geq E[D_3]$. Note that if U is NBU, this relationship is again reversed.

D. Method M_5

From the last two subsections, we learned that M_4 was Δ -consistent with respect to $G_U(x)$, while M_3 was biased unless U was exponential or the sampling rate was impossible (i.e., zero or infinity). However, a significant amount of previous effort went into measurement of the Internet using M_3 [6], [10], [26], [37], [41]. This raises the question of whether an existing collection of M_3 samples could be processed to remove the bias. To this end, define:

$$G_5(x_n, T) := \frac{1}{T} \sum_{j=1}^{T/\Delta} \min(x_n, A_U^+(s_j)) Q_{j,j+1} \quad (49)$$

to be an estimator that takes samples of M_3 , i.e., $A_U^+(s_j)$ conditioned on $Q_{j,j+1} = 1$, passes them through the min function, and normalizes the resulting sum by window size T . Note that the number of terms in the summation is $K(\infty, T)$, i.e., the number of detected updates.

Theorem 10: Estimator M_5 is Δ -consistent with respect to the age distribution:

$$G_5(x_n) := \lim_{T \rightarrow \infty} G_5(x_n, T) = G_U(x_n). \quad (50)$$

Proof: We start with an auxiliary result:

$$\begin{aligned} \sum_{k=0}^{n-1} \mathbf{1}_{A_U(s_j) > x_k} &= \sum_{k=0}^{n-1} \mathbf{1}_{A_U^+(s_j) > x_k} = \sum_{k=0}^{n-1} \mathbf{1}_{\lceil A_U(s_j) \Delta \rceil > k} \\ &= \min(n, \lceil A_U(s_j) \Delta \rceil) \\ &= \frac{\min(x_n, A_U^+(s_j))}{\Delta}. \end{aligned} \quad (51)$$

Next, applying this to expansion of (49):

$$\begin{aligned} G_5(x_n, T) &= \frac{\Delta}{T} \sum_{j=1}^{T/\Delta} Q_{j,j+1} \sum_{k=0}^{n-1} \mathbf{1}_{A_U(s_j) > x_k} \\ &= \frac{\Delta}{T} \sum_{k=0}^{n-1} \sum_{j=1}^{T/\Delta} \mathbf{1}_{A_U(s_j) > x_k} Q_{j,j+1} \\ &= \frac{K(\infty, T)}{N_S(T)} \sum_{k=0}^{n-1} \bar{G}_3(x_n, T), \end{aligned} \quad (52)$$

where $K(x, T)$ is given by (10) and $\bar{G}_3(x_n, T)$ by (41). Since $K(\infty, T)/N_S(T)$ converges to p , we get after applying (40) to $\bar{G}_3(x_n, T)$:

$$G_5(x_n) = p \frac{G_U(x_n)}{G_U(\Delta)} = G_U(x_n),$$

where we use the fact that $p = G_U(\Delta)$. ■

Fig. 13 shows that M_5 accurately obtains the tail of $G_U(x)$, even for Δ bounded away from zero. We next compare M_5 with M_4 to see if the reduction in the number of samples has a noticeable impact on accuracy. The first metric under consideration is the *Weighted Mean Relative Difference* (WMRD), often used in networking [15]. Assuming $H(x, T)$ is some empirical CDF computed in $[0, T]$, the WMRD between $H(x, T)$ and $G_U(x)$ is:

$$w_T := \frac{\sum_n |H(x_n, T) - G_U(x_n)|}{\sum_n (H(x_n, T) + G_U(x_n))/2}. \quad (53)$$

The second metric is the Kolmogorov-Smirnov (KS) statistic, which is the maximum distance between two distributions:

$$\kappa_T := \sup_{1 \leq n \leq T/\Delta} |H(x_n, T) - G_U(x_n)|. \quad (54)$$

Simulations results of (53)-(54) are shown in Table I. Observe that M_4 performs slightly better for $T \leq 10^3$, but then the two methods become identical.

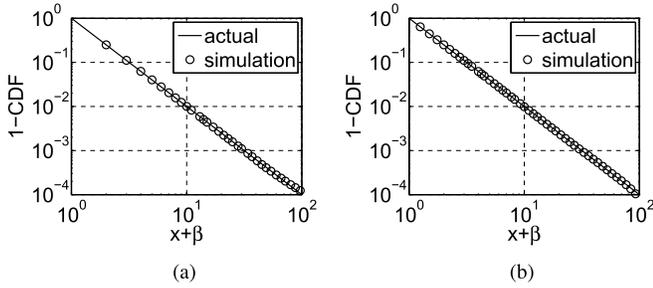


Fig. 13. Verification of (50) under Pareto U ($\mu = 2$). (a) $\lambda = 1$. (b) $\lambda = 4$.

TABLE I
CONVERGENCE OF BOTH Δ -CONSISTENT METHODS
UNDER PARETO U ($\mu = 2, \lambda = 1$)

T	M_4		M_5	
	w_T	κ_T	w_T	κ_T
10^2	3.48%	6.42%	3.72%	6.74%
10^3	1.37%	2.22%	1.40%	2.23%
10^4	0.47%	0.72%	0.47%	0.73%
10^5	0.15%	0.24%	0.15%	0.24%
10^6	0.04%	0.06%	0.04%	0.06%
10^7	0.02%	0.03%	0.02%	0.03%

VI. COMPARISON SAMPLING: RANDOM INTERVALS

Although M_4 and M_5 are consistent estimators of $G_U(x)$, they do not generally guarantee recovery of $F_U(x)$. Furthermore, constant S may not always be achievable in practice. For instance, search engine juggle trillions of pages, whose download rate is dynamically adjusted based on real-time ranking and budgeting. It may thus be difficult to ensure constant return delays to each page. Additional problems stem from lattice update processes, where constant S fails to satisfy Assumption 2, rendering measurements arbitrarily inaccurate.

In this section, we consider comparison sampling with random intervals. We first show that extending M_4 to this scenario delivers surprisingly biased results. Then, we present our new method M_6 and verify its correctness using simulations.

A. Straightforward Approaches

Our first attempt is to generalize M_4 to random S , which we call G- M_4 . For a given s_j , define the most-recent sample point after which an update has been detected as:

$$s_j^* := \max_{k < j} \{s_k : Q_{kj} = 1\}. \quad (55)$$

Then, G- M_4 rounds age $A_U(s_j)$ up to $s_j - s_j^*$. An example is shown in Fig. 14, where the measured value is $s_j - s_{j-2}$. For constant S , this method is identical to M_4 , which we know is consistent. The main difference with random S is that the amount of round-off error in G- M_4 varies from interval to interval. This issue has a profound impact on the result, as shown in Fig. 15. Observe that the exponential case becomes somewhat consistent only for $x_n \gg 0$ and the Pareto case produces a tail that is completely different from the actual $\tilde{G}_U(x)$. This motivates us to search for another approach.

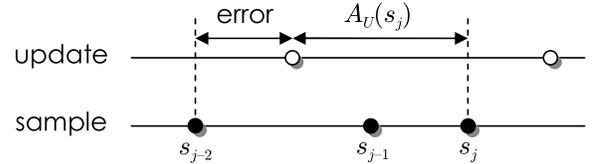


Fig. 14. Illustration of G- M_4 .

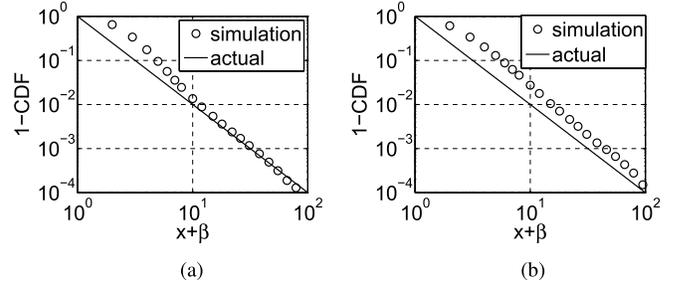


Fig. 15. Bias of G- M_4 with Pareto updates ($\mu = 2, \lambda = 1$). (a) exponential S . (b) Pareto S .

B. Method M_6

Our rationale for this technique stems from the fact that $Q_{ij} = 1$ if and only if $A_U(s_j) < s_j - s_i$. Therefore, counting the fraction of pairs (i, j) that sustain an update may lead to $G_U(x)$. Define $y^\circ = h \lceil y/h \rceil$ to be the rounded-up value of y with respect to a user-defined constant h . Let $y_n = nh$ and:

$$W_{ij}(y_n) := \begin{cases} 1 & (s_j - s_i)^\circ = y_n \\ 0 & \text{otherwise} \end{cases}. \quad (56)$$

Then, the number of inter-sample distances $s_j - s_i$ in $[0, T]$ that round up to y_n is given by:

$$W(y_n, T) := \sum_{i=1}^{N_S(T)} \sum_{j=i+1}^{N_S(T)} W_{ij}(y_n) \quad (57)$$

and the number of them with an update is:

$$Z(y_n, T) := \sum_{i=1}^{N_S(T)} \sum_{j=i+1}^{N_S(T)} Q_{ij} W_{ij}(y_n). \quad (58)$$

We can now define estimator M_6 by its CDF:

$$G_6(y_n, T) := \frac{Z(y_n, T)}{W(y_n, T)} \quad (59)$$

For a given λ , method M_6 has the same network overhead as the other methods; however, it utilizes $\Theta(n^2)$ pairwise comparisons, significantly more than the other methods, which are all linear in n . Despite a higher computational cost, M_6 gains significant accuracy advantages when distances $s_i - s_j$ are allowed to sweep all possible points $x \geq 0$. Combining this with bins of sufficiently small size creates a continuous CDF, which allows recovery of not only $G_U(x)$, but also $F_U(x)$.

Theorem 11: Assume $h \rightarrow 0$, N_S is age-mixing, and $F_S(x) > 0$ for all $x > 0$. Then, method M_6 is consistent

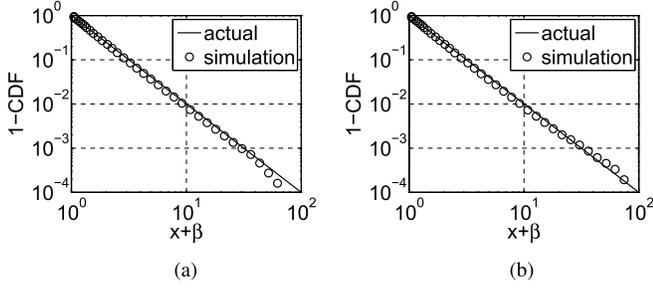


Fig. 16. Simulations of M_6 under Pareto updates ($h = 0.05, \mu = 2, \lambda = 1$). (a) exponential S . (b) Pareto S .

with respect to the age distribution:

$$G_6(y) := \lim_{T \rightarrow \infty} G_6(y, T) = G_U(y). \quad (60)$$

Proof: First, it helps to observe that:

$$Q_{ij} = \mathbf{1}_{R_U(s_i) \leq (s_j - s_i)}. \quad (61)$$

Since the download process is renewal, it follows that:

$$s_j - s_i \sim F_S^{*(j-i)}, \quad (62)$$

where $F_S^{*k}(x)$ denotes a k -fold convolution of distribution $F(x)$. Furthermore, the renewal nature of N_S implies that variable $s_j - s_i$ is independent of s_i . Now, let

$$Y_k \sim F_S^{*k}(x) \quad (63)$$

be a random variable with the same distribution as $S_1 + \dots + S_k$ and define the renewal function driven by $F_S(x)$ as [47]:

$$M_S(t) = 1 + \sum_{k=1}^{\infty} F_S^{*k}(t). \quad (64)$$

Then, renewal theory shows for $x > h$ and $n \rightarrow \infty$ that:

$$\frac{1}{n} \sum_{i=1}^n \sum_{j=i+1}^n \mathbf{1}_{R_U(s_i) \leq (s_j - s_i)} \mathbf{1}_{s_j - s_i \in (x-h, x]} \quad (65)$$

converges to

$$\begin{aligned} & \sum_{k=1}^{\infty} P(R_U \leq Y_k, Y_k \in (x-h, x]) \\ &= \sum_{k=1}^{\infty} \int_{x-h}^x G_U(y) dF_S^{*k}(y) = \int_{x-h}^x G_U(y) dM_S(y). \end{aligned} \quad (66)$$

Let $n = N_S(T)$ and assume that $h(T) = T^{-\delta}$, where $\delta \in (0, 1)$ ensures that h diminishes to zero at some appropriate rate. Since $G_U(x)$ is continuous, it follows that:

$$\lim_{T \rightarrow \infty} \frac{Z(y_n, T)}{W(y_n, T)} = \lim_{h \rightarrow 0} \frac{\int_{x-h}^x G_U(y) dM_S(y)}{\int_{x-h}^x dM_S(y)} = G_U(x) \quad (67)$$

for each $x > 0$. ■

The assumption that $F_S(x)$ contains non-zero mass in the vicinity of zero is necessary for accurate estimation of $g_U(x)$ at $x = 0$, which then leads to $F_U(x)$. This can be accomplished by a number of continuous distributions,

TABLE II
CONVERGENCE OF M_2 AND M_6 UNDER PARETO U AND EXPONENTIAL S ($\mu = 2, \lambda = 1$)

T	M_2		M_6	
	w_T	κ_T	w_T	κ_T
10^2	3.77%	7.56%	9.23%	33.17%
10^3	1.49%	2.64%	2.99%	7.86%
10^4	0.49%	0.79%	0.90%	2.30%
10^5	0.15%	0.29%	0.31%	0.93%
10^6	0.04%	0.07%	0.11%	0.38%
10^7	0.02%	0.03%	0.06%	0.13%

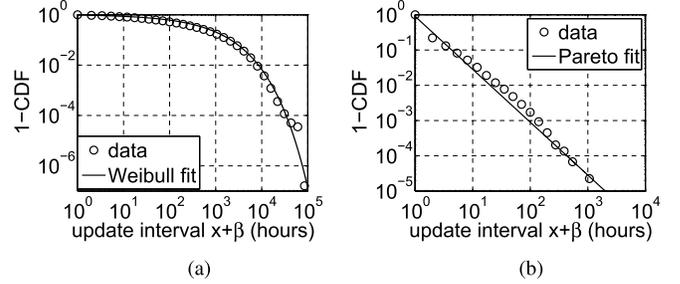


Fig. 17. Wikipedia inter-update delay distribution $F_U(x)$. (a) all articles. (b) George W. Bush.

e.g., $\exp(\lambda)$ or uniform in $[0, 2/\lambda]$. It should also be noted that M_6 can work for constant S , but in that case it offers no benefits over M_4 . Fig. 16 compares the M_6 estimator of $G_U(x)$ under two sampling distributions $F_S(x)$, both satisfying Theorem 11. Compared to Fig. 15, this result is overwhelmingly better.

The error of M_6 is contrasted against that of M_2 in Table II using exponential S and different interval lengths T . While the former is indeed converging slower than the latter, this was expected. Also notice that M_2 performs slightly worse with random S than with constant in Table I.

VII. DISCUSSION

Our remaining task in the paper is to compare the proposed methods on real traces and provide a recommendation of which approach to employ in what situation. We also take this opportunity to examine the Poisson assumption and how often it may hold under user-driven update processes.

A. Poisson Assumption

Wikipedia [45] is one of the most frequently visited websites on the Internet, with over 10B hits per month and 26M users participating in collaborative editing of encyclopedia-style articles. It can be argued that the properties of its user-driven update process is not only realistic, but also similar to those in other Internet scenarios (e.g., Facebook and Twitter posts, congestion updates in Google maps, news reports). What makes Wikipedia interesting is that it offers dumps of all modification timestamps $\{u_i\}$ across the entire collection of pages. Unlike other studies [4], [6], [11], [21], [27], [37], [41], [44], which relied on web or P2P crawling, usage of Wikipedia allows reconstruction of the ground-truth update process N_U .

TABLE III
TOP 10 MOST MODIFIED WIKIPEDIA ARTICLES

Page	Description	Updates	$E[U]$ (hours)	Lifetime T (years)
1	George W. Bush	44,296	1.86	9.4
2	LWWE*	33,744	1.46	5.6
3	Wikipedia	30,407	2.70	9.4
4	United States	28,771	2.90	9.5
5	Michael Jackson	25,558	3.22	9.4
6	Jesus	23,027	3.82	10.1
7	Britney Spears	21,549	3.85	9.5
8	World War II	21,424	3.83	9.4
9	Wii	21,023	3.00	7.2
10	Adolf Hitler	20,964	3.91	9.6

*List of World Wrestling Entertainment, Inc. personnel

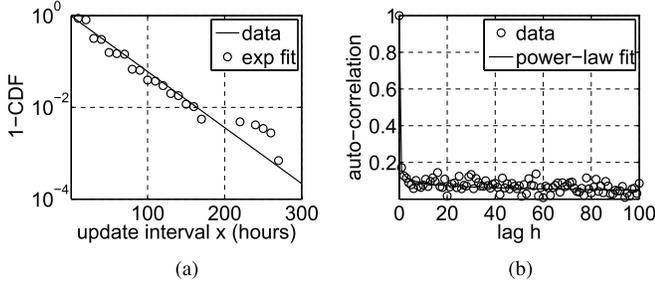


Fig. 18. Wikipedia pages with approximately exponential $F_U(x)$. (a) distribution. (b) correlation $\rho(h)$.

We use March 2011 traces of $n = 3.6$ million English Wikipedia articles [45] and compute inter-update intervals $\{U_{ij}\}$ across all pages i and all updates j . Fig. 17(a) displays the tail distribution of the aggregate result. Curve fitting on the data shows a good match to Weibull tail $e^{-(x/\nu)^k}$ with $\nu = 400$ and $k = 0.5$. This plot is as a mixture of n update distributions, or equivalently the expected $F_U(x)$ for a random page pulled from the Wikipedia database, where selection is made with a bias proportional to the number of updates in each page. If articles are downloaded by users (and thus need to be sampled) in correlation with their modification rate, the mixture CDF sheds light on the expected update dynamics faced by a sampling process. Needless to say, it is far from Poisson.

Focusing on individual pages, it is not surprising that the average number of updates per article is low (i.e., 71), making the corresponding $F_U(x)$ uninteresting for the majority of pages. However, the top 10 most-modified articles have a rich history (i.e., 6 – 10 years) and boast over 20K updates each, as shown in Table III. As their distribution $F_U(x)$ is similarly heavy-tailed, we show $\bar{F}_U(x)$ only for the #1 page “George W. Bush.” The result in Fig. 17(b) fits a Pareto tail $(1 + x/\beta)^{-\alpha}$ with $\alpha = 1.4$ and $\beta = 0.93$ pretty well.

To further investigate existence of pages that are modified by Poisson processes, we use a well-known fact that exponential U exhibits coefficient of variation $v = \sqrt{\text{Var}[U]}/E[U]$ equal to 1. First, we extract all pages with at least 100 updates and obtain 444K results. Among these, we seek articles with $|v - 1| \leq 0.1$, which yields a mere 202 pages. As shown in Fig. 18(a) for one of them, the tail of $F_U(x)$ matches exponential pretty well. However, to establish a Poisson N_U ,

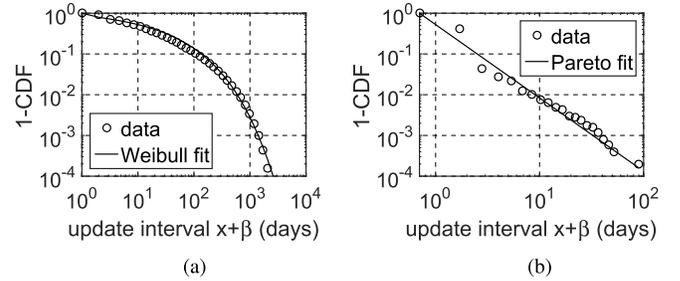


Fig. 19. Yelp inter-update delay distribution $F_U(x)$. (a) all businesses. (b) most-reviewed business.

TABLE IV
UNBIASED-METHOD COMPARISON ON WIKIPEDIA USING w_T ($\lambda = 2$)

Page	Exponential S		Constant S			
	M_2	M_6	M_2	M_4	M_5	M_6
1	0.14%	0.35%	0.07%	0.07%	0.08%	0.08%
2	0.06%	0.16%	0.02%	0.02%	0.02%	0.02%
3	0.11%	0.34%	0.05%	0.05%	0.05%	0.05%
4	0.10%	0.34%	0.04%	0.04%	0.04%	0.04%
5	0.10%	0.42%	0.06%	0.06%	0.06%	0.06%
6	0.12%	0.42%	0.05%	0.05%	0.05%	0.05%
7	0.16%	0.47%	0.06%	0.06%	0.06%	0.06%
8	0.13%	0.38%	0.04%	0.04%	0.04%	0.04%
9	0.15%	0.51%	0.07%	0.07%	0.07%	0.07%
10	0.13%	0.40%	0.04%	0.04%	0.04%	0.04%

we must additionally verify independence between updates. To this end, we plot the page’s auto-correlation function $\rho(h)$ in Fig. 18(b), which has a power-law trend $h^{-0.25}$. This suggests long-range dependence (LRD) with Hurst parameter 0.87, which is incompatible with the Poisson assumption. The remaining 201 pages produce analogous conclusions.

Zipf and Pareto dynamics have been known to emerge in many areas (e.g., web and AS graphs [5], [17], Internet traffic [34], peer lifetimes [41], [44], citation networks [3]). One of the theories [2] for this phenomenon lies in the bursty (i.e., ON/OFF) behavior of human influence on various data structures and traffic. It is therefore reasonable that social interaction between users and flash-crowd activity in response to events is likely to produce heavy-tailed $F_U(x)$ in a variety of Internet systems outside Wikipedia. One such example is the Yelp public-challenge dataset [49] that contains 2.2M reviews of 77K anonymized businesses. Its top-10 list has an order of magnitude fewer updates than the corresponding values in Table III and timestamp granularity is 1 day instead of 1 second; however, despite these differences, Fig. 19 shows that Yelp’s qualitative results (i.e., Weibull and Pareto distributions) are remarkably similar to those in Fig. 17.

B. Method Comparison

Since all consistent methods first estimate $G_U(x)$ and then apply numerical derivative to obtain $F_U(x)$, it suffices to assess performance using $G_U(x)$. We sample each page in Table III using $E[S] = 0.5$ hours (i.e., $\lambda = 2$) for its entire lifetime T . We first consider all unbiased methods. For M_2 , as well as M_6 with age-mixing S , we compute the residual CDF using bins of size $h = 0.05$ hours (i.e., 3 minutes). For M_4 - M_5 , as

TABLE V
UNBIASED-METHOD COMPARISON ON WIKIPEDIA USING κ_T ($\lambda = 2$)

Page	Exponential S		Constant S			
	M_2	M_6	M_2	M_4	M_5	M_6
1	0.28%	1.43%	0.20%	1.15%	1.15%	1.15%
2	0.34%	1.55%	0.25%	1.24%	1.24%	1.24%
3	0.26%	1.36%	0.15%	0.58%	0.58%	0.58%
4	0.23%	1.32%	0.13%	0.77%	0.77%	0.77%
5	0.22%	1.40%	0.12%	0.67%	0.67%	0.67%
6	0.21%	1.38%	0.10%	0.56%	0.56%	0.56%
7	0.22%	1.45%	0.10%	0.44%	0.44%	0.44%
8	0.24%	1.35%	0.10%	0.51%	0.51%	0.51%
9	0.26%	1.62%	0.14%	0.77%	0.77%	0.77%
10	0.26%	1.33%	0.09%	0.48%	0.48%	0.48%

TABLE VI
BIASED-METHOD COMPARISON ON WIKIPEDIA ($\lambda = 2$)

Page	Exponential S		Constant S			
	M_1		M_1		M_3	
	w_T	κ_T	w_T	κ_T	w_T	κ_T
1	6.0%	43.6%	6.0%	45.1%	6.0%	45.1%
2	0.4%	25.9%	0.4%	27.8%	3.6%	27.8%
3	4.7%	41.4%	4.7%	43.3%	5.3%	43.3%
4	7.9%	39.7%	8.0%	41.1%	8.0%	41.1%
5	17.5%	43.4%	17.6%	44.8%	17.6%	44.8%
6	18.5%	42.6%	18.5%	44.6%	18.5%	44.6%
7	13.8%	44.2%	13.9%	45.3%	13.9%	45.3%
8	4.1%	41.0%	4.1%	42.8%	6.6%	42.8%
9	6.7%	46.1%	6.8%	47.9%	6.8%	47.9%
10	9.8%	39.3%	9.8%	40.8%	9.8%	40.8%

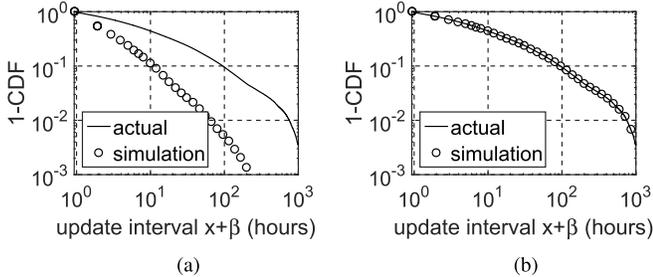


Fig. 20. Residual distribution $G_U(x)$ for George W. Bush. (a) M_3 . (b) M_4 .

well as M_6 with constant S , we linearly interpolate the step-function $G_U(x_n)$ to the granularity of h by adding 9 extra points between each x_n and x_{n+1} . Both cases limit CDF error computation to $x \in [0, 1000]$ hours.

Tables IV-V show the result. For exponential S , both M_2 and M_6 are viable options; however, the former achieves significantly lower error due to the higher accuracy of the information it receives (i.e., age). Although M_6 is consistent, its convergence speed is lower, which is consistent with our previous findings in Table II. For constant S , the tables show that M_2 again easily beats the other alternatives, with M_4 emerging in second place due to its simplicity. For biased methods, the corresponding numbers are given in Table VI. The error is not only high, but also insensitive to availability of age and variance of S .

To visualize the difference between prior methods and those derived in this paper, we offer graphical comparison in Figs. 20-21 using M_3 and M_4 as representatives of each class. We keep the sampling rate $\lambda = 2$ for Wikipedia (i.e., $E[S] = 30$ minutes) and $\lambda = 1$ for Yelp (i.e., $E[S] = 1$ day). Combined with Tables IV-VI, this leaves no doubt that the proposed framework achieves a significant improvement.

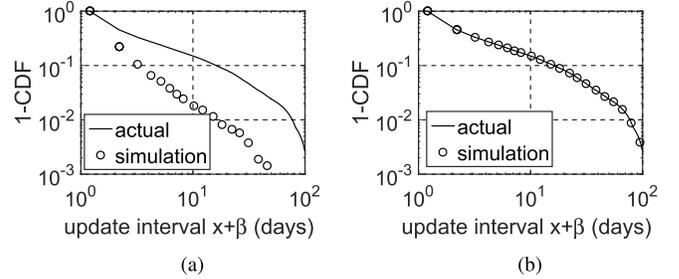


Fig. 21. Residual distribution $G_U(x)$ for the most-reviewed Yelp business. (a) M_3 . (b) M_4 .

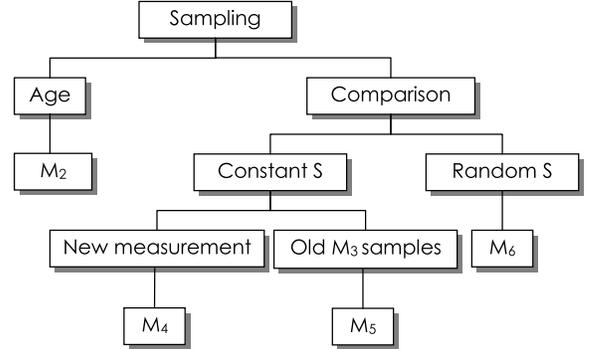


Fig. 22. Choosing among the proposed methods.

Summarizing the observations above, we have the final recommendation in Fig. 22. When age is available, M_2 converges the fastest and is always preferable. For comparison-based methods under constant S , method M_4 should be used for new measurements, while M_5 can be deployed to correct the bias of existing traces collected by M_3 . Finally, when age is unavailable and S is random, M_6 is the only option among those considered here.

VIII. CONCLUSION

This paper studied the problem of estimating the update distribution at a remote source under blind sampling. We analyzed prior approaches in this area, showed them to be biased under general conditions, introduced novel modeling techniques for handling these types of problems, and proposed several unbiased algorithms that tackled network sampling under a variety of assumptions on the information provided by the server and conditions at the observer. Simulations demonstrated that the introduced methods were significantly better than the existing state of the art in this field.

Future work includes derivation of convergence speed, investigation of non-parametric smoothing techniques for density estimation, and EM-based iterative estimation of the update distribution.

REFERENCES

- [1] E. Adar, J. Teevan, S. T. Dumais, and J. L. Elsas, "The Web changes everything: Understanding the dynamics of Web content," in *Proc. WSDM*, Jun. 2009, pp. 282–291.
- [2] A.-L. Barabási, "The origin of bursts and heavy tails in human dynamics," *Nature*, vol. 435, pp. 207–211, May 2005.
- [3] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, Oct. 1999.
- [4] B. E. Brewington and G. Cybenko, "How dynamic is the Web?" *Comput. Netw., Int. J. Comput. Telecommun. Netw.*, vol. 33, nos. 1–6, pp. 257–276, Jun. 2000.

- [5] A. Broder *et al.*, "Graph structure in the Web," *Comput. Netw., Int. J. Comput. Telecommun. Netw.*, vol. 33, pp. 309–320, Jun. 2000.
- [6] F. E. Bustamante and Y. Qiao, "Friendships that last: Peer lifespan and its role in P2P protocols," in *Web Content Caching and Distribution*. Dordrecht, The Netherlands: Springer, Sep. 2003, pp. 233–246.
- [7] J. Cho and H. Garcia-Molina, "The evolution of the Web and implications for an incremental crawler," in *Proc. VLDB*, Sep. 2000, pp. 200–209.
- [8] J. Cho and H. Garcia-Molina, "Synchronizing a database to improve freshness," in *Proc. ACM SIGMOD*, May 2000, pp. 117–128.
- [9] J. Cho and H. Garcia-Molina, "Effective page refresh policies for Web crawlers," *ACM Trans. Database Syst.*, vol. 28, no. 4, pp. 390–426, Dec. 2003.
- [10] J. Cho and H. Garcia-Molina, "Estimating frequency of change," *ACM Trans. Internet Technol.*, vol. 3, no. 3, pp. 256–290, Aug. 2003.
- [11] G. L. Ciampaglia and A. Vancheri, "Empirical analysis of user participation in online communities: The case of Wikipedia," in *Proc. ICWSM*, Sep. 2010, pp. 219–222.
- [12] E. Cohen and H. Kaplan, "The age penalty and its effect on cache performance," in *Proc. USENIX USITS*, Mar. 2001, pp. 73–84.
- [13] E. Cohen and H. Kaplan, "Refreshment policies for Web content caches," in *Proc. IEEE INFOCOM*, Apr. 2001, pp. 1398–1406.
- [14] D. Denev, A. Mazeika, M. Spaniol, and G. Weikum, "SHARC: Framework for quality-conscious Web archiving," in *Proc. VLDB*, Aug. 2009, pp. 183–207.
- [15] N. Duffield, C. Lund, and M. Thorup, "Estimating flow distributions from sampled flow statistics," in *Proc. ACM SIGCOMM*, Aug. 2003, pp. 325–336.
- [16] J. Edwards, K. McCurley, and J. Tomlin, "An adaptive model for optimizing performance of an incremental Web crawler," in *Proc. WWW*, May 2001, pp. 106–113.
- [17] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the Internet topology," in *Proc. ACM SIGCOMM*, Aug. 1999, pp. 251–262.
- [18] D. Fetterly, M. Manasse, and M. Najork, "On the evolution of clusters of near-duplicate Web pages," in *Proc. LA-WEB*, Nov. 2003, pp. 37–45.
- [19] C. Grimes and D. Ford, "Estimation of Web page change rates," in *Proc. ASA Joint Statist. Meetings (JSM)*, 2008.
- [20] C. Grimes, D. Ford, and E. Tassone, "Keeping a search engine index fresh: Risk and optimality in estimating refresh rates for Web pages," in *Proc. INTERFACE*, May 2008.
- [21] D. Gruhl *et al.*, "How to build a WebFountain: An architecture for very large-scale text analytics," *IBM Syst. J.*, vol. 43, no. 1, pp. 64–77, 2004.
- [22] R. Horincar, B. Amann, and T. Artières, "Online change estimation models for dynamic Web resources: A case-study of RSS feed refresh strategies," in *Proc. ICWE*, Jul. 2012, pp. 395–410.
- [23] J.-J. Lee, K.-Y. Whang, B. S. Lee, and J.-W. Chang, "An update-risk based approach to TTL estimation in Web caching," in *Proc. IEEE WISE*, Dec. 2002, pp. 21–29.
- [24] X. Li, D. B. H. Cline, and D. Loguinov, "On sample-path staleness in lazy data replication," in *Proc. IEEE INFOCOM*, Apr. 2015, pp. 1104–1112.
- [25] X. Li and D. Loguinov, "Stochastic models of pull-based data replication in P2P systems," in *Proc. IEEE P2P*, Sep. 2014, pp. 1–10.
- [26] N. Matloff, "Estimation of Internet file-access/modification rates from indirect data," *ACM Trans. Model. Comput. Simul.*, vol. 15, pp. 233–253, Jul. 2005.
- [27] A. Ntoulas, J. Cho, and C. Olston, "What's new on the Web?: The evolution of the Web from a search engine perspective," in *Proc. WWW*, May 2004, pp. 1–12.
- [28] M. Oita and P. Senellart, "Deriving dynamics of Web pages: A survey," in *Proc. TAWW*, Mar. 2011, pp. 25–32.
- [29] C. Olston and S. Pandey, "Recrawl scheduling based on information longevity," in *Proc. WWW*, Apr. 2008, pp. 437–446.
- [30] V. N. Padmanabhan and L. Qiu, "The content and access dynamics of a busy Web site: Findings and implications," in *Proc. ACM SIGCOMM*, Aug. 2000, pp. 111–123.
- [31] S. Pandey, K. Dhamdhere, and C. Olston, "WIC: A general-purpose algorithm for monitoring Web information sources," in *Proc. VLDB*, Aug. 2004, pp. 360–371.
- [32] S. Pandey and C. Olston, "User-centric Web crawling," in *Proc. WWW*, May 2005, pp. 401–411.
- [33] S. Pandey, K. Ramamritham, and S. Chakrabarti, "Monitoring the dynamic Web to respond to continuous queries," in *Proc. WWW*, May 2003, pp. 659–668.
- [34] V. Paxson and S. Floyd, "Wide area traffic: The failure of Poisson modeling," *IEEE/ACM Trans. Netw.*, vol. 3, no. 3, pp. 226–244, Jun. 1995.
- [35] K. Radinsky and P. N. Bennett, "Predicting content change on the Web," in *Proc. WSDM*, Feb. 2013, pp. 415–424.
- [36] D. Roselli, J. R. Lorch, and T. E. Anderson, "A comparison of file system workloads," in *Proc. USENIX Annu. Tech. Conf.*, Jun. 2000, pp. 41–54.
- [37] S. Saroiu, P. K. Gummadi, and S. D. Gribble, "Measurement study of peer-to-peer file sharing systems," *Proc. SPIE*, vol. 4673, pp. 156–170, Jan. 2002.
- [38] K. C. Sia, J. Cho, and H.-K. Cho, "Efficient monitoring algorithm for fast news alerts," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 7, pp. 950–961, Jul. 2007.
- [39] S. R. Singh, "Estimating the rate of Web page updates," in *Proc. Int. Joint Conf. Artif. Intell.*, Jun. 2007, pp. 2874–2879.
- [40] M. Spaniol, D. Denev, A. Mazeika, G. Weikum, and P. Senellart, "Data quality in Web archiving," in *Proc. WICOW*, Apr. 2009, pp. 19–26.
- [41] D. Stutzbach and R. Rejaie, "Understanding churn in peer-to-peer networks," in *Proc. ACM IMC*, Oct. 2006, pp. 189–202.
- [42] Q. Tan and P. Mitra, "Clustering-based incremental Web crawling," *ACM Trans. Inf. Syst.*, vol. 28, no. 4, Nov. 2010, Art. no. 17.
- [43] C. Voglis, P. E. Hadjidoukas, I. E. Lagaris, and D. G. Papageorgiou, "A numerical differentiation library exploiting parallel architectures," *Comput. Phys. Commun.*, vol. 180, no. 8, pp. 1404–1415, Aug. 2009.
- [44] X. Wang, Z. Yao, and D. Loguinov, "Residual-based estimation of peer and link lifetimes in P2P networks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 3, pp. 726–739, Jun. 2009.
- [45] (Jul. 2014). *Wikipedia Dumps*. [Online]. Available: <http://dumps.wikimedia.org/enwiki/20110317/>
- [46] J. L. Wolf, M. S. Squillante, P. S. Yu, J. Sethuraman, and L. Ozsen, "Optimal crawling strategies for Web search engines," in *Proc. WWW*, May 2002, pp. 136–147.
- [47] R. W. Wolff, *Stochastic Modeling and the Theory of Queues*. Upper Saddle River, NJ, USA: Prentice-Hall, 1989.
- [48] M. Yang, H. Wang, L. Lim, and M. Wang, "Optimizing content freshness of relations extracted from the Web using keyword search," in *Proc. ACM SIGMOD*, Jun. 2010, pp. 819–830.
- [49] (2016). *Yelp Dataset Challenge*. [Online]. Available: https://www.yelp.com/dataset_challenge



Xiaoyong Li (S'14) received the B.S. degree in computer engineering and the M.S. degree in electrical engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2005 and 2008, respectively. He is currently pursuing the Ph.D. degree in computer science with Texas A&M University, College Station.

His research interests include peer-to-peer systems, information retrieval, and stochastic analysis of networks.



Daren B. H. Cline received the B.S. degree in mathematics from Harvey Mudd College, Claremont, CA, USA, in 1978, and the M.S. and Ph.D. degrees in statistics from Colorado State University, Fort Collins, CO, USA, in 1980 and 1983, respectively. He joined the Statistics Department, Texas A&M University, College Station, TX, USA, in 1984, where he is currently a Tenured Professor. His current research interests include applied stochastic processes, Markov chain theory, heavy-tailed distributions, and nonlinear time series models.



Dmitri Loguinov (S'99–M'03–SM'08) received the B.S. (Hons.) degree in computer science from Moscow State University, Russia, in 1995, and the Ph.D. degree in computer science from the City University of New York, New York, in 2002.

He is currently a Professor with the Department of Computer Science and Engineering, Texas A&M University, College Station. His research interests include P2P networks, information retrieval, congestion control, Internet measurement, and modeling.