

# Residual-Based Estimation of Peer and Link Lifetimes in P2P Networks

Xiaoming Wang, *Student Member, IEEE*, Zhongmei Yao, *Student Member, IEEE*, and Dmitri Loguinov, *Senior Member, IEEE*

**Abstract**—Existing methods of measuring lifetimes in P2P systems usually rely on the so-called *Create-Based Method* (CBM), which divides a given observation window into two halves and samples users “created” in the first half every  $\Delta$  time units until they die or the observation period ends. Despite its frequent use, this approach has no rigorous accuracy or overhead analysis in the literature. To shed more light on its performance, we first derive a model for CBM and show that small window size or large  $\Delta$  may lead to highly inaccurate lifetime distributions. We then show that create-based sampling exhibits an inherent tradeoff between overhead and accuracy, which does not allow any fundamental improvement to the method. Instead, we propose a completely different approach for sampling user dynamics that keeps track of only *residual* lifetimes of peers and uses a simple renewal-process model to recover the actual lifetimes from the observed residuals. Our analysis indicates that for reasonably large systems, the proposed method can reduce bandwidth consumption by several orders of magnitude compared to prior approaches while simultaneously achieving higher accuracy. We finish the paper by implementing a two-tier Gnutella network crawler equipped with the proposed sampling method and obtain the distribution of ultrapeer lifetimes in a network of 6.4 million users and 60 million links. Our experimental results show that ultrapeer lifetimes are Pareto with shape  $\alpha \approx 1.1$ ; however, link lifetimes exhibit much lighter tails with  $\alpha \approx 1.8$ .

**Index Terms**—Gnutella networks, lifetime estimation, peer-to-peer, residual sampling.

## I. INTRODUCTION

PEER-TO-PEER networks are popular platforms for many applications such as file-sharing, content distribution, and multimedia streaming. Besides modeling and simulating system dynamics of P2P networks under churn (e.g., [4], [8], [10], [13]), validation of proposed techniques in real networks has recently become an important area for understanding P2P performance and design limitations in practice. In this regard, several efforts have been undertaken to characterize peer-to-peer systems by measuring churn-related user behavior (e.g., distribution of lifetime, inter-arrival delays, and availability) [1], [2], [5], [7], [18], [20], topological information (e.g., degree distribution and clustering coefficients) [14], [22], and traffic flow rate [10], [19].

Manuscript received April 26, 2007; revised December 07, 2007 and March 11, 2008; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor K. Ross. First published October 03, 2008; current version published June 17, 2009. This work was supported by the National Science Foundation (NSF) under Grants CCR-0306246, ANI-0312461, CNS-0434940, CNS-0519442, and CNS-0720571. An earlier version of this paper appeared in IEEE INFOCOM 2007, Anchorage, AK.

The authors are with the Department of Computer Science, Texas A&M University, College Station, TX 77843-3112 USA (e-mail: xmwang@cs.tamu.edu; mayyao@cs.tamu.edu; dmitri@cs.tamu.edu).

Digital Object Identifier 10.1109/TNET.2008.2001727

Sampling of large-scale networks usually faces two fundamental problems—1) obtaining an unbiased distribution of the target quantity and 2) keeping bandwidth overhead reasonable as system size increases. While sampling bias in topology measurement is understood fairly well [21], the same issue in lifetime sampling has not been addressed before. What makes the latter problem different is that sampled users *cannot* be queried for their lifetimes or even arrival instances. Measurement in such cases generally requires taking repeated snapshots of the system every  $\Delta$  time units, detecting new arrivals by user appearance in a given snapshot, and inferring departures based on user absence in another snapshot. Since  $\Delta$  cannot be lowered below the delay it takes to crawl the network, the issue of precisely reconstructing the lifetime distribution from measured samples remains open.

In this paper, we aim to formalize the notion of lifetime sampling bias, understand its source in existing methods, and design a robust and bandwidth-efficient sampling mechanism for estimating peer and link lifetime distributions in unstructured P2P networks (e.g., Gnutella [9], KaZaA [11]). Note that peer lifetimes are important for understanding general user behavior, their habits, and application performance offered by the peers to the system. Link lifetimes, on the other hand, have a significant impact on resilience [13], [23] and routing ability [12] of the network since broken links, rather than dead peers, contribute to formation of stale neighbor pointers, network disconnection, and routing failure.<sup>1</sup>

We start by creating a novel analytical framework for understanding and characterizing bias in lifetime sampling. We first explain what constitutes inaccuracy in measuring the target distribution of lifetimes  $F(x)$  and define sampling methods to be *biased* if, given an infinite population of sampled users, they cannot reproduce  $F(x)$  in all discrete points  $j\Delta$  in the interval  $[\Delta, T]$ . Armed with this definition, we then offer a closed-form model for the measurements obtained by *Create-Based Method* (CBM) [17], which is a widely used heuristic for sampling lifetimes in computer systems. We show that both CBM and its modification in [2], [18], [20] are generally biased as long as  $\Delta > 0$ , where the bias is caused by two factors—inconsistent round-offs (i.e., some user lifetimes are rounded up and others down) and missed users (i.e., users arrive and depart within a  $\Delta$  interval). In fact, we generalize this result to show that *any* sampling technique that attempts to directly measure user lifetimes every  $\Delta$  time units is biased as long as  $\Delta > 0$  and that the bias

<sup>1</sup>There are many reasons why peer lifetime may be different from link lifetime, which include peers reaching their maximum neighbor capacity and dropping excess links, leaves migrating from one ultrapeer to another to achieve better performance, path outages between certain nodes, and demotion of ultra-peers to leaf status.

is not removable regardless of the mathematical manipulation applied to the measured samples.

To overcome the discovered limitations of direct sampling, we next propose a technique called *Residual-based Estimator* (RIDE), in which a crawler takes a snapshot of the entire network and then tracks the residual (i.e., remaining) lifetimes of the users seen in the first crawl. We show that this approach produces an unbiased version of the residual distribution  $H(x)$ , which allows us to develop a simple mechanism based on renewal churn models of [13], [23] that accurately reconstructs the lifetime distribution  $F(x)$  from the sampled residuals with a negligible amount of error.

The next issue we address is bandwidth consumption of lifetime sampling. With small  $\Delta$  and large  $T$ , CBM requires significant overhead since it must track *all* users that appear in the system in the observation interval, i.e., old peers discovered early in the crawl and new ones constantly arriving into the system.<sup>2</sup> In RIDE, however, initial users die quickly and the amount of bandwidth needed to sustain the crawl decays to zero proportionally to the tail of the residual lifetime distribution  $H(x)$ . Additional bandwidth savings are possible if the initial set  $S_0$  of users found in the system is uniformly subsampled and only  $\epsilon$ -fraction of the users is monitored during the interval  $[0, T]$ . For example, given Pareto lifetimes with  $\alpha = 1.1$  observed in our experiments, window  $T = 24$  hours, and sampling interval  $\Delta = 3$  minutes, the proposed technique reduces the download overhead compared to that in CBM by a factor of 16 for  $\epsilon = 0.1$  and a factor of 125 for  $\epsilon = 0.01$ .

We finish the paper by implementing a Gnutella crawler that is about 18 times faster than the fastest prior crawler [20], which allows it to cover the entire network of 6.4 million users (1.2 million contacted ultrapeers) in under 3 minutes. Our results using RIDE indicate that ultrapeer lifetimes are Pareto distributed with shape  $\alpha \approx 1.1$ , which is very close to the results of [2]. At the same time, Gnutella links are much more volatile and can be described by a Pareto distribution with shape  $\alpha \approx 1.8$ . These results, fed into the latest resilience models for unstructured systems [13], [23], suggest that node isolation among joining ultrapeers in Gnutella and thus partitioning of the network must indeed be extremely rare events.

The remainder of the paper is organized as follows. In Section II, we formalize sampling and bias. In Section III, we derive the sampling bias of CBM and examine it under different simulation settings. We propose the residual-based method and discuss its simulation results in Section IV. We analyze the subsampling technique in Section V, examine the bandwidth overhead of the various methods in Section VI and present our measurement study of Gnutella in Section VII. Section VIII reviews prior work and Section IX concludes the paper.

## II. FORMALIZING LIFETIME SAMPLING

### A. Target Distribution

We start by defining the objective of our measurement process. Assume that each user spends a random amount of

<sup>2</sup>Note that besides lifetimes CBM can measure additional metrics (e.g., arrival/departure process of users) and its overhead might be justified when these metrics are important.

time in the system, where the lifetime  $L$  of joining users is drawn from some distribution  $F(x)$ . This is similar to the heterogeneous churn model proposed in [23]. Then, the goal of the sampling process is to estimate with as much accuracy as possible function  $F(x)$ , which we assume is continuous *almost everywhere*<sup>3</sup> in the interval  $(0, \infty)$ . As shown in [23], distribution  $F(x)$  represents the lifetimes of *arriving* rather than *existing* peers in the system. The latter metric is known in renewal process theory as the *spread* of user lifetimes and can be obtained from  $F(x)$  using simple integration.

The measurement process is assumed to have periodic access to the information about which users are currently present in the system. This process allows the sampler to test whether a given user  $i$  is still alive as well as discover the entire population of the system at any time  $t$ . However, due to bandwidth and connection-delay constraints on obtaining this information, the sampling process cannot query the system for longer than  $T$  or more frequently than once per  $\Delta$  time units, where  $\Delta$  usually varies from several minutes to several hours depending on the speed of the crawler and network size.

Given the above requirements, notice that reconstructing the entire  $F(x)$  from discrete samples is simply impossible. There are three biases arising from discrete sampling: 1) the measuring process cannot observe any lifetimes larger than  $T$ ; 2) all samples are rounded to a multiple of  $\Delta$ ; 3) an empirical distribution based on a finite sample size will not necessarily match the theoretical one. We are not concerned with the last issue since *all* methods require an infinitely large sample size to converge to the desired distribution  $F(x)$ . Instead, we are interested in the bias arising from finite  $T$  and non-zero  $\Delta$ .

We start with the following definition that formalizes samples obtained during periodic measurements.

*Definition 1:* A non-negative random variable  $X^\Delta$  for some  $\Delta > 0$  is called *lattice* if

$$\sum_{j=1}^{\infty} P(X^\Delta = j\Delta) = 1 \quad (1)$$

where  $\Delta$  is called the *periodicity* of  $X^\Delta$  and points  $x_j = j\Delta$  are called the *support* of  $X^\Delta$ .

For all lattice distributions, we assume that  $P(X^\Delta \leq 0) = F(0) = 0$  and that the probability mass of  $X^\Delta$  starts from the point  $x_1 = \Delta$ .

We are now ready to define a sampling process.

*Definition 2:* A  $(\Delta, T)$ -*sampling process* is a lattice random variable  $M^\Delta$  with periodicity  $\Delta$  and  $P(\Delta \leq M^\Delta \leq T) = 1$ .

Note that the above defines a sampling process using the *limiting* distribution of the values it measures (i.e., assuming an infinite population size). The reason for doing so is to understand whether a method can provide accurate results given a sufficiently large sampling size. As we show below, some methods *always* exhibit bias, no matter how long they measure the system.

*Definition 3:* For a random variable  $X$ , function  $E(x)$  is called an *estimator* of  $X$  in some interval  $[a, b]$  if it is the CDF of some random variable  $Y$  that approximates  $X$  in  $[a, b]$ .

<sup>3</sup>The set of points in which  $F(x)$  is discontinuous must have measure 0.

Note that  $Y$  can be arbitrarily dissimilar to  $X$ , in which case the estimator will be biased. We next explain what makes an estimator unbiased.

**Definition 4:** A  $(\Delta, T)$ -sampling process with estimator  $E(x)$  is *unbiased* with respect to a target continuous random variable  $X$  if it can correctly reproduce the distribution of  $X$  in all discrete points  $x_j$  in the interval  $[\Delta, T]$  for any  $\Delta > 0$ :

$$E(x_j) = P(X \leq x_j) \quad (2)$$

for  $x_j = j\Delta$  and  $j = 1, 2, \dots, T/\Delta$ .

Since one may measure different aspects of the system, we finally classify sampling methods based on whether they measure the target random variable or some other related distribution.

**Definition 5:** A  $(\Delta, T)$ -sampling process of a random variable  $X$  is called *direct*, if it measures quantities whose distribution is the same as that of  $X$ . It is called *indirect* otherwise.

For example, direct lifetime sampling must measure session lengths of all arriving users, while indirect sampling may record the lifetimes of peers alive in the system at some time  $t$ . Given an established relationship between the two metrics, an estimator can then be used to reconstruct lifetimes  $L$  from indirect samples. In another example, direct sampling of network size must count the number of users present in the system at different times  $t$ , while indirect sampling may measure the arrival process of peers into the system. Properly selected indirect sampling may be more accurate and/or may require lower overhead than direct sampling. We demonstrate one such example later in the paper.

### III. DIRECT SAMPLING

In this section, we first examine the source of bias in direct sampling and study the problem of constructing an unbiased estimator for measuring lifetimes. We then derive a model for the distribution obtained by Create-Based Method (CBM) and demonstrate examples of its bias.

#### A. General Results

In direct sampling, the measured random variable  $M^\Delta$  is the lifetime of individual users conditioned on them being smaller than  $T$  and being present in the crawl:

$$P(M^\Delta \leq x) = P(L \leq x | L \leq T, \text{ not missed}) \quad (3)$$

where missed samples arise when a user joins and departs between consequent crawls. Note, however, that not all users with lifetimes smaller than  $\Delta$  are missed and that some of them are actually taken into account in the distribution of  $M^\Delta$ . Another issue that we discover in this work is that some lifetime samples are rounded up and others rounded down during the measurement, which together with missed users gives rise to the bias we derive below. We next formalize round-off errors and explain how they affect direct sampling.

**Definition 6:** For a continuous random variable  $X$ , a  $(\Delta, T)$  sampling process is *consistent* if measured samples are all rounded up to the nearest multiple of  $\Delta$ .

Since a crawler in direct sampling never knows the exact arrival time of users it observes, there is an ambiguity in how to round-off the lifetimes of measured peers. Consider the example

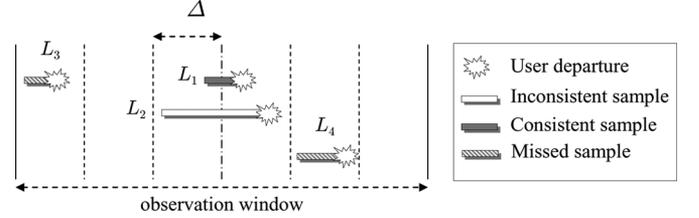


Fig. 1. Round-off inconsistencies in direct sampling.

in Fig. 1, where sample  $L_1 = 0.5\Delta$  is indistinguishable from sample  $L_2 = 1.8\Delta$  from the perspective of the crawler. This causes both of these lifetimes to be rounded off to  $\Delta$ , which using our terminology makes  $L_1$  consistent and  $L_2$  inconsistent. Also observe in the figure that samples  $L_3 = 0.4\Delta$  and  $L_4 = 0.6\Delta$  are completely missed by the crawler, even though sample  $L_1$  is captured. This case can also be treated as inconsistent round-off as we define below.

Let

$$Q_j = \begin{cases} 1 & \text{inconsistently rounded down to } x_j \\ 0 & \text{otherwise} \end{cases}$$

to be an indicator variable of the event that a user's lifetime  $x_j \leq L < x_{j+1}$  is inconsistently rounded down to  $x_j$  by the sampling process, where rounding down to  $x_0 = 0$  represents missing the entire sample. For simplicity of notation, we define  $\rho_j = P(Q_j = 1)$  and obtain the probability of inconsistent round-off in the interval  $[x_j, x_{j+1})$  in the next theorem.

**Theorem 1:** In direct sampling, the probability that lifetime samples are inconsistently rounded down to  $x_j = j\Delta$  ( $j = 0, 1, \dots, T/\Delta$ ) is:

$$\rho_j = \frac{1}{\Delta} \int_{x_j}^{x_{j+1}} F(x) dx - F(x_j) \quad (4)$$

where  $F(x)$  is the CDF of the lifetime distribution of samples.

*Proof:* Without loss of generality, shift the time axis such that a given user arrives at time  $0 \leq t < \Delta$  into the system and its lifetime is  $x_j < L \leq x_{j+1}$ . Then, the user is sampled inconsistently with probability:

$$g(t) = P(x_j < L \leq x_{j+1} - t) = F(x_{j+1} - t) - F(x_j). \quad (5)$$

Since arrival point  $t$  can be any uniformly random point in  $[0, \Delta]$ , we have:

$$\rho_j = \int_0^\Delta g(t) \frac{1}{\Delta} dt \quad (6)$$

which leads to the desired result.  $\blacksquare$

Equipped with (4), we next derive an unbiased estimator for the continuous random variable  $L$ .

**Theorem 2:** For direct lifetime sampling, the following is an unbiased estimator of  $L$ :

$$E(x_j) = P(M^\Delta \leq x_j) \times P(L \leq T | Q_0 = 0)(1 - \rho_0) + \rho_0 - \rho_j, \quad (7)$$

where  $\rho_j$  is given in (4).

*Proof:* Note that for a measured lifetime sample  $M^\Delta \leq x_j$ , either the actual lifetime  $L \leq x_j$  holds, or  $x_j < L <$

$x_{j+1}$  but is inconsistently rounded down to  $x_j$ , i.e.,  $Q_j = 1$ . Therefore, (3) becomes

$$P(M^\Delta \leq x_j) = P(L \leq x_j \cup Q_j = 1 | L \leq T, Q_0 = 0). \quad (8)$$

Denoting by  $B$  the event  $(L \leq x_j \cup Q_j = 1)$  and expanding the conditional probability on the right side of (8), we get

$$\begin{aligned} P(M^\Delta \leq x_j) &= \frac{P(B, L \leq T, Q_0 = 0)}{P(L \leq T, Q_0 = 0)} \\ &= \frac{P(B, Q_0 = 0)}{P(L \leq T, Q_0 = 0)}, \end{aligned} \quad (9)$$

where the second equality is from the fact that  $B$  implies  $L \leq T$ . Next, we derive the numerator and denominator of the right-hand side of (9) separately.

Rewrite the numerator of (9) as follows:

$$P(B, Q_0 = 0) = P(B) - P(B, Q_0 = 1), \quad (10)$$

where  $P(B)$  can be expanded by

$$P(B) = P(L \leq x_j) + P(Q_j = 1) = P(L \leq x_j) + \rho_j. \quad (11)$$

Since  $Q_0 = 1$  implies  $L \leq x_j$ , we have

$$P(B, Q_0 = 1) = P(Q_0 = 1) = \rho_0. \quad (12)$$

Substituting (11)–(12) into (10), we obtain

$$P(B, Q_0 = 0) = P(L \leq x_j) + \rho_j - \rho_0. \quad (13)$$

Expanding the denominator of (9) establishes that

$$\begin{aligned} P(L \leq T, Q_0 = 0) &= P(L \leq T | Q_0 = 0)P(Q_0 = 0) \\ &= P(L \leq T | Q_0 = 0)(1 - \rho_0). \end{aligned} \quad (14)$$

Substituting (13)–(14) into (9) gives

$$P(M^\Delta \leq x_j) = \frac{P(L \leq x_j) + \rho_j - \rho_0}{P(L \leq T | Q_0 = 0)(1 - \rho_0)} \quad (15)$$

Considering that an unbiased estimator  $E(x_j)$  should equal  $P(L \leq x_j)$ , (7) follows from (15). ■

We next verify (4) and (7) in simulations using the example of a Pareto distribution commonly used to model user lifetimes [13], [23]:

$$F(x) = 1 - (1 + x/\beta)^{-\alpha}, \quad \alpha > 1, x \geq 0 \quad (16)$$

with  $E[L] = \beta/(\alpha - 1)$ . We use  $\alpha = 1.1$ ,  $\beta = 0.05$ ,  $T = 24$  hours, and  $E[L] = 0.5$  hours. We count the number of inconsistent round-offs for each  $j = 0, 1, \dots, T/\Delta$  and plot the corresponding empirical probability  $\rho_j$  in Fig. 2, which shows that (4) predicts reality very well. Furthermore, we compute the empirical values of  $P(M^\Delta \leq x_j)$  and supplement the measured data with the knowledge of  $\rho_j$  to obtain  $E(x_j)$  according to (7). Fig. 3 plots the values of  $E(x_j)$  obtained both from model (7) and the actual distribution (16), which indicates a perfect match.

From the result of Theorem 2, it becomes clear that unbiased measurement requires access to the distribution of observed

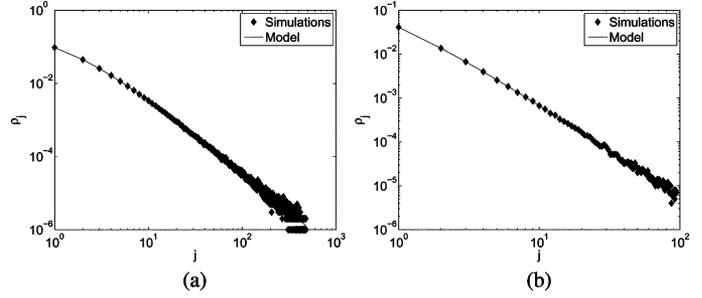


Fig. 2. Comparison of  $\rho_j$  computed from (4) to simulations. (a)  $\Delta = 3$  min. (b)  $\Delta = 15$  min.

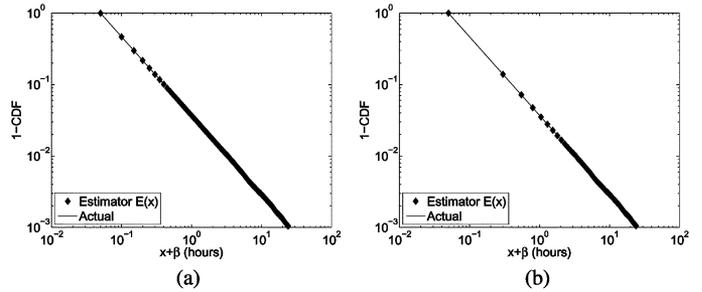


Fig. 3. Verifying estimator (7) in simulations. (a)  $\Delta = 3$  min. (b)  $\Delta = 15$  min.

samples (i.e., variable  $M^\Delta$ ), the fraction of observed lifetimes that are no larger than  $T$  (i.e.,  $P(L \leq T | Q_0 = 0)$ ), and all individual  $\rho_j$ . While the first two metrics are easily measurable in practice, recovery from inconsistent round-offs requires the exact join time of each sampled user and the number of missed users. Unfortunately, within the constraints of our problem (i.e., crawling of alive users with a period no less than  $\Delta$ ), the effect of round-off errors is impossible to overcome no matter what manipulation is applied to  $M^\Delta$ .

### B. Create-Based Method (CBM)

We next study how inconsistent round-offs exhibit themselves in a widely used [2], [18], [20] direct sampling algorithm called *Create-Based Method* (CBM), first introduced by [17] in the context of operating systems. Recall from [17] that CBM uses an observation window of size  $2T$ , which is split into small intervals of size  $\Delta$ . Within the observation window  $[0, 2T]$ , the algorithm takes a snapshot of the system at the beginning of each interval. To avoid sampling bias, [17] suggests dividing the window into two halves and only including samples that appear during the first half of the window, disappear somewhere within the window, and stay in the system no longer than  $T$  time units. Fig. 4 shows an example of create-based sampling with three valid, four invalid, and two missed lifetime samples. The invalid cases include users who join the system before the observation window or in its second half, a peer that survives beyond time  $2T$ , and a user whose lifetime is larger than  $T$ .

Assume that  $N$  is the number of users that arrive into the system in the first half of the window  $[0, T]$  and  $N(x)$  is the number of such users with lifetimes less than or equal to  $x$ . Observe that  $N(T)$  is the number of valid samples collected by CBM and  $\lim_{N \rightarrow \infty} N(T)/N$  is the simply metric  $P(L \leq$

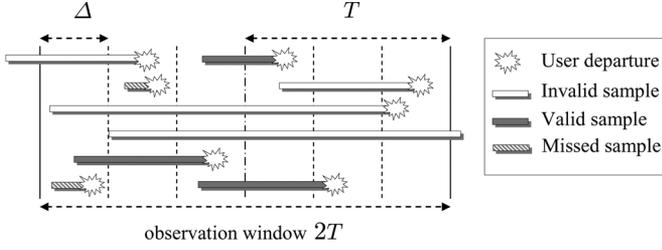


Fig. 4. Illustration of sampling in CBM.

$T|Q_0 = 0)$  defined earlier. One possible way to estimate  $F(x)$  is to take the ratio of  $N(x_j)$  to  $N(T)$  as the estimator of the probability  $P(L \leq x_j)$ , which leads to our first CBM estimator [17]:

$$E_A(x_j) = \lim_{N \rightarrow \infty} \frac{N(x_j)}{N(T)} = P(M^\Delta \leq x_j). \quad (17)$$

Recent work in [2], [18], [20] normalizes  $E_A$  by the percentage of samples no larger than  $T$  (i.e.,  $N(T)/N$ ) and defines the following modified estimator:

$$E_B(x_j) = \lim_{N \rightarrow \infty} \frac{N(x_j)}{N}. \quad (18)$$

With the result in (7), we can express both CBM estimators as functions of the actual distribution  $F(x_j) = P(L \leq x_j)$ .

*Theorem 3:* Both CBM estimators (17)–(18) are generally biased when any  $\rho_j > 0$  and produce the following distributions:

$$E_A(x_j) = \frac{F(x_j) - \rho_0 + \rho_j}{F(T) - \rho_0}, \quad E_B(x_j) = \frac{F(x_j) - \rho_0 + \rho_j}{1 - \rho_0}. \quad (19)$$

*Proof:* Expanding  $E_A(x_j)$  using (7), we have:

$$E_A(x_j) = \frac{F(x_j) - \rho_0 + \rho_j}{P(L \leq T|Q = 0)(1 - \rho_0)}. \quad (20)$$

We now use the following reasoning. Observe that  $P(L \leq T|Q = 0)P(Q = 0) = P(L \leq T, Q = 0)$ , which can be rewritten as follows:

$$\begin{aligned} P(L \leq T, Q = 0) &= P(L \leq T) - P(L \leq T, Q = 1) \\ &= P(L \leq T) - \rho_0. \end{aligned} \quad (21)$$

The second equality in (21) is from the fact that  $Q = 1$  implies  $L \leq T$ . The result of  $E_B(x_j)$  is a direct consequence of (7). ■

The result in [19] shows that  $E_B$  is closer to  $F(x)$  than  $E_A$  since its accuracy is not affected by the value of  $T$ . Next, we explore in more detail the effect of  $(\Delta, T)$  on the fidelity of these estimators using model [19] and simulations.

### C. Effect of Bias on CBM

We first explain how  $T$  and  $\Delta$  skew the shape of estimator  $E_A$ . To simplify the discussion below, define  $\bar{E}(x) = 1 - E(x)$  to be the tail distribution of any CDF function  $E(x)$ . It then follows from [19] that:

$$\bar{E}_A(x_j) = \frac{\bar{F}(x_j) - \bar{F}(T) - \rho_j}{F(T) - \rho_0} \quad (22)$$

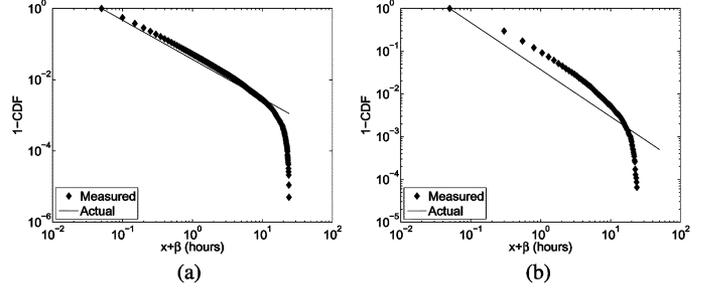


Fig. 5. Estimator  $E_A$  with Pareto lifetimes ( $n = 10^6$  users,  $T = 24$  hours,  $\alpha = 1.1$ ,  $\beta = 0.05$ , and  $E[L] = 0.5$  hours). (a)  $\Delta = 3$  min. (b)  $\Delta = 15$  min.

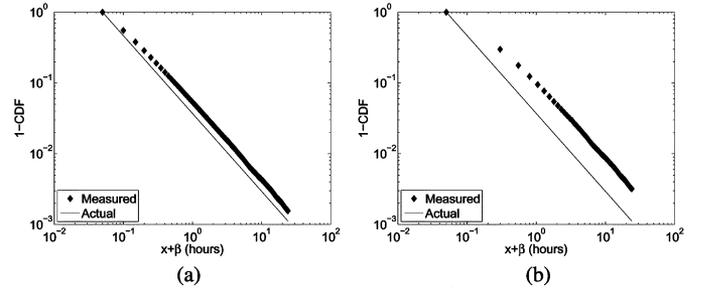


Fig. 6. Estimator  $E_B$  with Pareto lifetimes ( $n = 10^6$  users,  $T = 24$  hours,  $\alpha = 1.1$ ,  $\beta = 0.05$ , and  $E[L] = 0.5$  hours). (a)  $\Delta = 3$  min. (b)  $\Delta = 15$  min.

which shows that the measured tail distribution is a shifted and scaled version of the true tail. The influence of the shift/scale factors on the right side of (22) could be illustrated through simulations. We use CBM with  $T = 24$  hours in a hypothetical network with  $n = 1$  million users that join and depart using the churn model of [23]. Even though  $F(T) = 99.8\%$  of users have lifetimes smaller than  $T$ , Fig. 5 shows that  $E_A$  suffers from significant bias that increases as  $\Delta$  becomes larger. Not only does the measured distribution  $E_A$  produce incorrect estimates  $\alpha \approx 2.4$ ,  $\beta \approx 0.5$  of Pareto parameters when fitted with the corresponding curve, but the shape of the tail in Fig. 5 does not even resemble that of  $\bar{F}(x)$ , which may lead to erroneous conclusions about the family of distributions  $F(x)$  belongs to.

We now study how  $\rho_j$  affects the shape of  $E_B$ . It follows from [19] that for  $j = 0, 1, 2, \dots, T/\Delta$ :

$$\bar{E}_B(x_j) = \frac{\bar{F}(x_j) - \rho_j}{1 - \rho_0} \quad (23)$$

which is the true tail shifted by  $\rho_j$  and then scaled by  $1 - \rho_0$ . For small  $\rho_j \approx 0$ , this transformation on log scale preserves the Pareto shape parameter  $\alpha$  as seen in Fig. 6, but makes scale parameter  $\beta$  inaccurate (i.e.,  $\alpha \approx 1.14$ ,  $\beta \approx 0.15$  for  $\Delta = 15$  minutes). For cases of non-negligible  $\rho_j$  that arise when  $\Delta$  is very large or when distribution  $F(x)$  does not admit shape invariance during scaling (e.g., Gaussian, uniform), estimator  $E_B$  may produce significantly misleading results.

### D. Limitations of CBM

From now on, we refer to  $E_B$  when mentioning CBM since  $E_B$  exhibits less bias than  $E_A$  as shown by model (19) and Figs. 5–6. We next investigate whether there exists a lifetime distribution such that  $\bar{E}_B(x_j) = \bar{F}(x_j)$ . Notice from (23) that

there are two ways for CBM to be unbiased, either  $\rho_0 = \rho_j = 0$  or  $\rho_j = \rho_0 \bar{F}(x_j)$ . The next two theorems shows that only special lifetime distributions satisfy these conditions.

*Theorem 4:* The only lifetime distribution that allows CBM to avoid all round-off errors (i.e.,  $\rho_j = 0, j = 0, 1, 2, \dots, T/\Delta - 1$ ) is a step function with support  $m\Delta$  for some integer  $m \geq 1$ .

*Proof:* It follows from (4) that for  $\rho_j = 0$  ( $j \geq 1$ ) to hold, we must have

$$F(x) = \begin{cases} 0 & 0 \leq x < \Delta \\ F(x_j) & x_j \leq x < x_j + \Delta, \end{cases} \quad (24)$$

which specifies a step function with support  $m\Delta$ , where  $m = 1, 2, \dots$  ■

Note that in real-life systems, the assumption that the lifetime distribution follows a step function with periodicity equal to crawl delay  $\Delta$  is too restrictive. Therefore, we next seek non-step functions that satisfy the second condition of CBM being unbiased.

*Theorem 5:* The only lifetime distribution that allows CBM to be unbiased simultaneously for all  $\Delta > 0$  is exponential.

*Proof:* We prove the theorem by first verifying that the exponential function satisfies  $\bar{E}_B(x_j) = \bar{F}(x_j)$  and show its uniqueness. Substituting  $F(x) = 1 - e^{-x/\mu}$  into (4), we obtain  $\rho_j$  for exponential distributions

$$\rho_j = e^{-x_j/\mu} \left( 1 - \frac{\mu}{\Delta} (1 - e^{\Delta/\mu}) \right) = \bar{F}(x_j) \rho_0. \quad (25)$$

Substituting (25) into (23) establishes  $\bar{E}_B(x_j) = \bar{F}(x_j)$ .

We next show that if  $\bar{E}_B(x_j) = \bar{F}(x_j)$ , then  $F(x)$  must be an exponential function. Assuming  $\bar{E}_B(x_j) = \bar{F}(x_j)$ , we can reduce (23) to:

$$\rho_j = \bar{F}(x_j) \rho_0. \quad (26)$$

Expanding  $\rho_j$  and  $\rho_0$  in (26) using (4), we get

$$\int_0^\Delta (F(x + x_j) - F(x_j)) dx = \int_0^\Delta \bar{F}(x_j) F(x) dx. \quad (27)$$

For (27) to hold for all  $\Delta > 0$ , we must have for all  $x > 0$

$$F(x + x_j) - F(x_j) = \bar{F}(x_j) F(x) \quad (28)$$

which can be simplified to

$$\bar{F}(x + x_j) = \bar{F}(x) \bar{F}(x_j). \quad (29)$$

Note that the only solution to  $\bar{F}(x + y) = \bar{F}(x) \bar{F}(y)$  is given by the exponential function  $\bar{F}(x) = e^{-x/\mu}$ , which establishes the desired result. ■

Given that the observed lifetimes in peer-to-peer systems [2], [20] are heavy-tailed, we next explore a different method that works without any assumptions on  $F(x)$ .

#### IV. INDIRECT SAMPLING

In this section, we seek a solution to the problem of achieving both high accuracy and low overhead using indirect sampling. It has been suggested [1], [13], [23] that users in peer-to-peer systems can be modeled as alternating between available (ON)

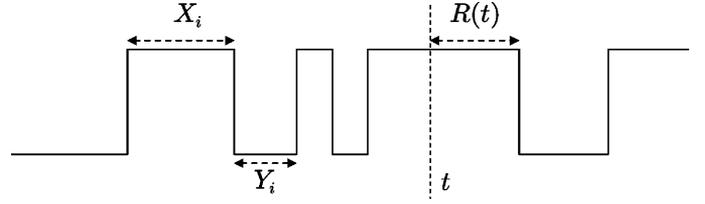


Fig. 7. Process  $Z_i(t)$  depicting user  $i$ 's ON/OFF behavior.

and unavailable (OFF) states. Inspired by these efforts, we now propose our measurement algorithm, called *Residual-based Estimator* (RIDE), that exploits renewal theory [15] to reconstruct  $F(x)$  from sampled residual lifetimes.

#### A. Churn Model

Consider a P2P system with  $n$  participating users, where each user  $i$  is either alive (i.e., present in the system) at time  $t$  or dead (i.e., logged off). This behavior can be modeled by an ON/OFF process  $\{Z_i(t)\}$  for each user  $i = 1, 2, \dots, n$ :

$$Z_i(t) = \begin{cases} 1 & \text{user } i \text{ is alive at time } t \\ 0 & \text{otherwise.} \end{cases} \quad (30)$$

This framework is illustrated in Fig. 7, where  $X_i$  and  $Y_i$  are i.i.d. durations of user  $i$ 's ON (life) and OFF (death) periods, respectively, and  $R(t)$  is the remaining lifetime of user  $i$  at time  $t$ . Assume that variables  $X_i$  are drawn from a user-specific distribution  $F_i(x) = P(X_i < x)$  and denote by  $l_i = E[X_i]$  the expected lifetime and by  $d_i = E[Y_i]$  the expected offtime of user  $i$ . Further define  $b_i$  to be the arrival rate of user  $i$  normalized by the total arrival rate into the system [23]:

$$b_i = \frac{1/(l_i + d_i)}{\sum_{k=1}^n 1/(l_k + d_k)}. \quad (31)$$

Then, it has been proven in [23] that the aggregate lifetime of all users seen by the system is drawn from a weighted distribution:

$$F(x) = \sum_{i=1}^n b_i F_i(x). \quad (32)$$

As before,  $L$  is the random lifetime of peers visiting the system and the goal of our and other measurement studies is not to sample each of  $F_i(x)$ , but rather to measure the users' aggregate behavior  $F(x) = P(L < x)$ . In order to accomplish this task, define  $R(t)$  be the residual lifetime of a random alive user at time  $t$  and denote by  $H(x)$  its limiting distribution:

$$H(x) = \lim_{t \rightarrow \infty} P(R(t) \leq x). \quad (33)$$

Then,  $F(x)$  can be inferred from  $H(x)$  using the following relationship established in [23]:

$$H(x) = \frac{1}{\mu} \int_0^x (1 - F(u)) du \quad (34)$$

where  $\mu = E[L]$  is the expected lifetime of a joining peer and  $F(x)$  is given by (32).

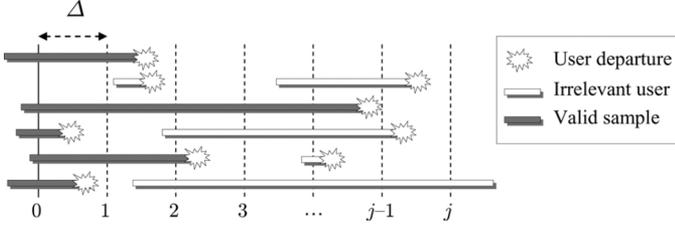


Fig. 8. Sampling residuals in RIDE.

## B. RIDE

We first define the sampling algorithm in RIDE and then discuss its estimator  $E_R(x)$ . At time  $t_0$ , RIDE takes a snapshot of the whole system and records in set  $S_0$  all users found to be alive. For all subsequent intervals  $j$  ( $j = 1, 2, \dots, T/\Delta$ ) of  $\Delta$  time units, the algorithm keeps probing peers in set  $S_0$  either until they die or  $T$  expires. After the observation window is over, the algorithm obtains the distribution of residual lifetime  $M^\Delta$  of the users in set  $S_0$ .

Two important properties about residual sampling can be drawn from its definition: 1) no valid samples can be missed since only users who are alive at time  $t = t_0$  are valid measurements; 2) no samples can be inconsistently rounded off since all valid residual lifetimes start from the time of the first crawl. Fig. 8 illustrates an example of five valid samples captured in the first crawl and five irrelevant lifetimes that are safely ignored by the algorithm.

Define  $E_H(x_j)$  to be an estimator of the residual distribution  $H(x)$  using users in set  $S_0$  and assuming the system is in equilibrium:

$$E_H(x_j) = \lim_{|S_0| \rightarrow \infty} \frac{N(x_j)}{|S_0|} \quad (35)$$

where  $N(x)$  denotes the number of users in  $S_0$  whose lifetimes are shorter than or equal to  $x$ . Since RIDE does not miss or incorrectly round off samples, its estimation of residual lifetimes is unbiased, which leads to  $E_H(x_j) = H(x_j)$ . Combining with (34), we establish the next theorem.

*Theorem 6:* For residual lifetime sampling, the following is an unbiased estimator of  $L$ :

$$E_R(x_j) = 1 - \frac{h(x_j)}{h(0)} \quad (36)$$

where  $x_j = j\Delta$  and  $h(x) = H'(x)$  is the PDF of  $R(t)$ . Furthermore, the expected user lifetime is  $E[L] = 1/h(0)$ .

*Proof:* Differentiating both sides of (34), we get  $h(x) = (1 - F(x))/\mu$ . Setting  $x = 0$ , it follows that  $h(0) = 1/\mu$ . Solving for  $F(x)$ , we get (36). ■

Since  $H(x)$  is computed without bias, it is now possible to numerically compute its derivative  $h(x)$  using Taylor expansion with error bounded by  $O(H^{(k)}(x)\Delta^k/k!)$ , where  $k = T/\Delta$  is the number of samples in the curve. For any analytic function  $H(x)$ , the convergence of the error to zero is guaranteed as  $k$  becomes large. For  $\Delta = 3$  minutes and  $T = 24$  hours commonly used in our experiments, the resulting error for Pareto lifetimes with  $\alpha = 1.1$  and  $\beta = 0.1$  is upper bounded by

$H^{(480)}(0)\Delta^{480}/480! \approx 10^{-624}$ , which for all practical purposes can be considered zero. In simulations, however, we find that using only 3 points is often sufficient for achieving good estimation accuracy (see below).

## V. SUBSAMPLING

In this section, we examine under what conditions CBM and RIDE allow reduction of measurement overhead through some type of user subsampling. We then discuss the algorithm used in RIDE and show its performance.

### A. Preliminaries

Assume that the target P2P system is fully distributed and supports the operation of building a list  $S$  of currently alive peers. We assume that it takes  $\Delta$  time units to create  $S$  and that this process requires bandwidth overhead proportional to the number of alive users, i.e.,  $C|S|$  bytes where  $C$  is some fixed bandwidth needed to find an alive user and download its parameters (e.g., IP address and port number). Note that we do not require that the P2P system provide any other mechanisms that aid our measurement process (e.g., notifications about user arrival, departure, or dynamic link changes).

Once  $S$  is built, we assume that each alive user can be monitored using a mechanism independent of the P2P network to detect its departure (e.g., using TCP connection requests). This monitoring also incurs overhead  $C$  units per user and can be done no more often than once per  $\Delta$  time units per peer in order to keep the process non-intrusive and scalable to large  $n$ . Decoupling lifetime probing from the initial collection of alive users, we allow for a range of subsampling techniques where only a subset of alive users is monitored at any given time. With such algorithms, the goal is to simultaneously reduce the total overhead in the observation window  $[0, T]$  and preserve estimation accuracy.

Our earlier description of CBM required repeated crawls of the system to refresh set  $S$ , detect new arrivals, and measure their lifetimes. With certain additional restrictions (see below), there are two possible mechanisms for subsampling in CBM. The first method, which we call *direct*, obtains the initial snapshot  $S_0$  of the system at time  $t_0$  and then selects each user with an independent probability  $\epsilon$  into a smaller subset  $S'_0$ . The measurement process then monitors the ON/OFF behavior of each user  $v \in S'_0$  for  $T$  time units and estimates  $F(x)$  based on the collected lifetimes (ignoring the first ON cycle of each user since it contains a residual rather than a lifetime). The second method, which we call *indirect*, monitors the *neighbors* of users in  $S'_0$  to detect new arrivals and uses their lifetimes for estimating the distribution of  $L$ .

RIDE subsampling is similar to the first CBM technique in the sense that only residuals of users in  $S'_0$  are monitored and used in estimation.

### B. Direct CBM Sampling

We start our analysis with direct subsampling. There are several restrictions that this method imposes: 1) users between sessions must appear with the same (IP, port) combination; 2) OFF durations are small in comparison to  $T$ ; and 3) the distribution of lifetimes for users in  $S'_0$  is indeed  $F(x)$ . In many current

P2P networks, the user is likely to use a new IP address assigned by its DHCP/PPP server [20] or choose a new port for each join into the system [9], which makes detection of its return into the system and monitoring of its ON/OFF cycles impossible without periodic re-crawls of the system. The second assumption depends on user behavior and may significantly impact CBM subsampling if users do not return into the system within  $T$  time units, which has been observed in BitTorrent networks [20]. Even if both conditions 1) and 2) are satisfied, our next result shows that even with  $\Delta = 0$  direct CBM subsampling is biased in general networks and does not converge to  $F(x)$ .

Assume that  $S_0$  is a random variable representing the set of currently alive users,  $N(x, T)$  is the average number of lifetimes less than  $x$  collected by CBM from users in  $S_0$  during  $[0, T]$ , and  $N(T)$  is the average number of samples observed by CBM in the same interval. Then, define a new CBM estimator that measures the lifetime of users in  $S_0$ :

$$E_{BS}(x) = \lim_{T \rightarrow \infty} \frac{N(x, T)}{N(T)}. \quad (37)$$

*Theorem 7:* For  $\Delta = 0$ , the distribution of user lifetimes in  $S_0$  is

$$E_{BS}(x) = \sum_{i=1}^n \frac{l_i/(l_i + d_i)^2}{\sum_{k=1}^n l_k/(l_k + d_k)^2} F_i(x). \quad (38)$$

*Proof:* Denote by  $\Omega(T)$  the set of all lifetime samples obtained from monitoring users in  $S_0$  in the interval  $[0, T]$ . We use average-case analysis where each user is included into  $S_0$  with probability  $l_i/(l_i + d_i)$ , i.e., with probability that it is alive at time  $t_0$ . It then follows that the expected fraction of samples from user  $i$  in  $\Omega(T)$  as  $T \rightarrow \infty$  is [23]

$$p_i = \frac{l_i/(l_i + d_i)^2}{\sum_{k=1}^n l_k/(l_k + d_k)^2}. \quad (39)$$

Therefore, the probability that a randomly selected lifetime in an infinite set  $\lim_{T \rightarrow \infty} \Omega(T)$  is from user  $i$  and less than  $x$  is  $p_i P(L_i < x)$ , which leads to

$$E_{BS}(x) = \sum_{i=1}^n p_i P(L_i < x) \quad (40)$$

and thus establishes (38).  $\blacksquare$

Since uniformly random subsampling of  $S_0$  produces users with the same bias as in the original set, we immediately obtain the next result.

*Corollary 1:* For sufficiently large  $\epsilon|S_0|$ , direct CBM subsampling estimates distribution (38).

Due to limited space, we omit simulations showing the accuracy of (38) and the extent of its deviation from  $F(x)$  in (32). Theorem 7 shows that, compared to peers arriving into the system, set  $S_0$  is more heavily skewed towards users with significant online presence. The key difference between CBM and RIDE is that the former requires observation of *new* arrivals into the system, while the latter needs to monitor users *currently* in the system. Therefore, RIDE naturally expects bias among users

in  $S_0$ , which it overcomes using a reconstruction technique in (36). CBM has no such mechanism.

Our final observation regarding direct CBM subsampling is that it can be used only in systems consisting of homogeneous peers (i.e., all users exhibit identical characteristics with  $F_i(x) = F(x)$ ). However, since measurement studies [20] show that P2P users are often heterogeneous, this assumption is of limited practical use.

### C. Indirect CBM Sampling

Indirect sampling imposes the following restrictions: 1) neighbors of alive users can be monitored using some P2P protocol; 2) arrivals into a certain neighbor set only include new users (i.e., peers do not dynamically switch links); 3) neighbors of peers in set  $S'_0$  have unbiased lifetimes; 4) set  $S'_0$  is popular enough to attract a sufficient number of new neighbors in  $[0, T]$ ; and 5) users in  $S'_0$  depart from the system slower than new users become their neighbors.

The first restriction is relatively easy to satisfy in certain systems (e.g., Gnutella), but may be more problematic in cases when the current population  $S$  is known to some server-like entity (e.g., BitTorrent tracker) in the form of a list, but individual peer-to-peer links are unknown. The second restriction does not hold even in Gnutella since users frequently migrate from one ultrapeer to another. These migrating users then arrive into existing neighbor sets with non-zero age, which creates bias in all cases except exponential  $L$ .

The third constraint depends on the neighbor selection algorithm and may introduce bias in lifetimes of observed arrivals if users make neighbor selection based on the characteristics of their future neighbors. Since set  $S_0$  is already biased from the perspective of CBM, arriving neighbors may also exhibit bias if links are formed based on some similarity between  $F_i(x)$  of new users and  $F_j(x)$  of existing peers. Without a specific graph-construction algorithm, further analysis of this bias is impossible. The last two restrictions also depend on how P2P graphs are constructed and may become a problem when new users for whatever reason do not choose peers in  $S'_0$  as neighbors or when  $|S'_0|$  shrinks to zero before enough new arrivals have been detected.

### D. RIDE Subsampling

The accuracy of RIDE's subsampling algorithm can be inferred from the fact that users in  $S'_0$  have the same distribution of residuals as those in  $S_0$ .

*Corollary 2:* For sufficiently large  $\epsilon|S_0|$ , RIDE subsampling produces an unbiased estimate of  $H(x)$ .

Compared to CBM, RIDE allows subsampling under the most general conditions and requires only two main assumptions introduced in the beginning of this section (i.e., ability to construct  $S_0$  and monitor alive users until they die). RIDE subsampling does not impose any restrictions on link structure, user migration, lifetime homogeneity, user appearance in subsequent sessions, OFF durations, or neighbor selection. As a result, for the most general case assumed throughout the paper, CBM must perform full crawls to build its estimates of  $F(x)$ , while RIDE can monitor residuals in  $S'_0$ .

It is worthwhile mentioning that residual sampling acquires all valid samples during the very first crawl. Therefore, given

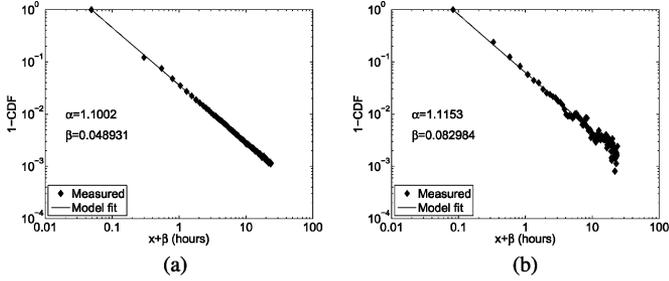


Fig. 9. Original and subsampled estimator  $E_R$  with Pareto lifetimes ( $|S_0| = 10^6$  users,  $T = 24$  hours,  $\Delta = 15$  minutes,  $\alpha = 1.1$ ,  $\beta = 0.05$ , and  $E[L] = 0.5$  hours). Both examples use 3-point derivatives. (a)  $\epsilon = 1$ . (b)  $\epsilon = 0.01$ .

that  $|S_0|$  is sufficiently large, Corollary 2 shows that it is possible to randomly subsample the initial set of users and track the residuals of only  $\epsilon$  percent of the entire user population. This significantly reduces traffic requirements and allows RIDE to achieve orders of magnitude lower bandwidth overhead in practice compared to CBM. We leave the overhead analysis of subsampling to the next section and now show in Fig. 9 one example of using this technique, where a system of 1 million users in the same setup as in Fig. 6(b) is subsampled by a factor of 100. First notice in Fig. 9(a) that RIDE recovers  $F(x)$  with much higher accuracy than  $E_B$  and obtains  $\alpha = 1.1002$  and  $\beta = 0.049$ . Second, observe in Fig. 9(b) that RIDE achieves reasonable estimation accuracy ( $\alpha = 1.115$ ,  $\beta = 0.083$ ) even with just 10 000 users; however, the tail of the subsampled distribution is highly variable, which potentially makes it difficult to understand the distribution's qualitative behavior. We next deal with this issue.

### E. Inverse Averaging

To overcome the tail noise arising when  $|S_0|$  is heavily subsampled, we next present an algorithm for reducing the variance in the measured distribution  $E_R(x)$ . Notice that  $E_R(x)$  is a mapping between two discrete sets, i.e., from set  $\mathcal{X} = \{j\Delta\}$  to set  $\mathcal{Y} = \{j/|S_0|\}$  for  $j = 1, 2, \dots, T/\Delta$ . For each  $y \in \mathcal{Y}$ , we find all  $x_i \in \mathcal{X}$  such that  $E_R(x_i) = y$  and calculate the corresponding average  $\hat{x}(y)$ :

$$\hat{x}(y) = \frac{\sum_i x_i \mathbf{1}_{E_R(x_i)=y}}{\sum_k \mathbf{1}_{E_R(x_k)=y}} \quad (41)$$

where  $\mathbf{1}_A$  is the indicator function of event  $A$ . Denote by  $\hat{\mathcal{X}}$  the set of all possible  $\hat{x}(y)$  from (41), i.e.,  $\hat{\mathcal{X}} = \{\hat{x}(y)|y \in \mathcal{Y}\}$  and define *inverse averaging* to be a relation  $(\hat{x}(y), y)$  for all  $y \in \mathcal{Y}$ . By smoothing out the tail, inverse averaging improves the shape of the distribution and allows better accuracy in estimation.

Next, we examine two cases of inverse averaging using the example in Fig. 9(a) subsampled with  $\epsilon = 0.1$  and  $\epsilon = 0.01$ . The resulting distributions are shown in Fig. 10, which demonstrates much better preservation of the Pareto shape in the tail and less oscillations than without the use of inverse averaging. For  $\epsilon|S_0| = 10^5$  in Fig. 10(a), curve fitting produces  $\alpha = 1.11$ ,  $\beta = 0.065$ , and for  $\epsilon|S_0| = 10^4$  in Fig. 10(b), we obtain  $\alpha = 1.09$ ,  $\beta = 0.077$ . This shows that even when  $\Delta$  is comparable to the average lifetime  $E[L]$  and with very few samples, RIDE is capable of reasonably accurate estimation.

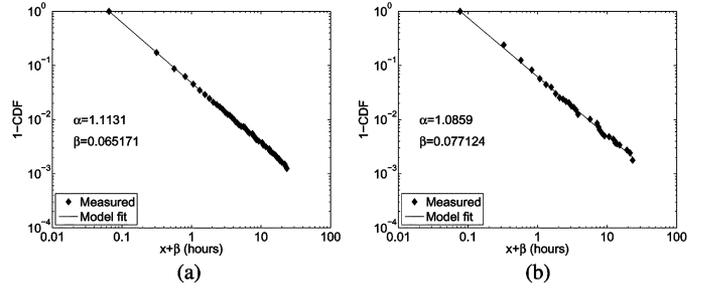


Fig. 10. Inverse averaging applied to  $E_R$  for Pareto lifetimes ( $|S_0| = 10^6$  users,  $T = 24$  hours,  $\Delta = 15$  minutes,  $\alpha = 1.1$ ,  $\beta = 0.05$ , and  $E[L] = 0.5$  hours). Both examples use 3-point derivatives. (a)  $\epsilon = 0.1$ . (b)  $\epsilon = 0.01$ .

### F. Evaluation

In this section, we provide comparison results between subsampled RIDE and original CBM that cover various parameter settings and lifetime distributions. We start with defining a statistical metric that characterizes the extent of difference between two distribution functions. The metric that we use in this section is the Weighted Mean Relative Difference (WMRD), which is often used for comparing heavy-tailed distributions [6]. Denote by  $F(x)$  the actual CDF function and by  $E(x)$  its estimator. Define  $\delta$  to be the WMRD distance between  $E(x)$  and  $F(x)$ , which is computed as

$$\delta = \frac{\sum_j |E(x_j) - F(x_j)|}{\sum_j (E(x_j) + F(x_j))/2} \quad (42)$$

where  $x_j = j\Delta$ . Note that the asymptotic value of  $\delta$  in CBM (i.e., for  $N \rightarrow \infty$ ) can be computed from [19] given any lifetime distribution and sampling parameters. It can also be inferred from [19] that CBM's limiting  $\delta$  is generally non-zero for non-lattice, non-exponential lifetimes. On the other hand, with  $\epsilon|S_0| \rightarrow \infty$ , RIDE's  $\delta$  converges to zero.

Setting asymptotics aside, we next examine CBM and RIDE using  $\delta$  in simulations with finite systems. Fig. 11 plots the values of  $\delta$  obtained from simulations with Pareto lifetimes. In the figure, we vary each of the four parameters while fixing the others according to the basic setting of  $T = 24$  hrs,  $\Delta = 3$  min,  $\alpha = 1.1$ , and  $E[L] = 30$  min. For each scenario, we simulate a measurement process that captures  $10^7$  samples, estimates the lifetime distribution using CBM and RIDE, and computes the resulting difference  $\delta$  between the estimated and actual distributions. The figure shows that RIDE even using 1% sub-sampling ( $\epsilon = 0.01$ ) exhibits smaller errors than the original CBM in all parameter configurations. Fig. 12 also compares CBM and RIDE with lifetimes drawn from the uniform distribution in  $(\mu - 0.5, \mu + 0.5)$  and the Weibull distribution with shape  $\alpha = 0.6$  and scale  $\beta = 0.66\mu$ , where  $\mu$  is the expected lifetime. The figure indicates that RIDE achieves similar and even better performance than CBM in all studied cases.

## VI. OVERHEAD

This section formalizes the overhead of the various studied sampling methods and compares the bandwidth requirement of RIDE to that of CBM.

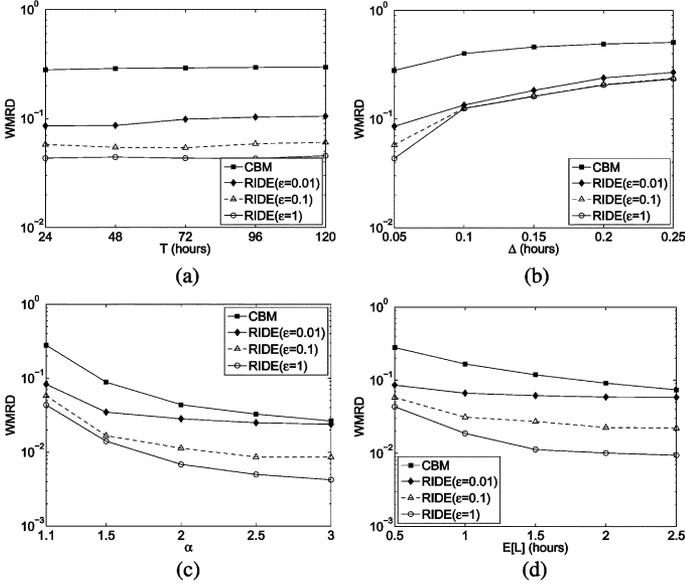


Fig. 11. Comparing CBM and RIDE for Pareto lifetimes ( $T = 24$  hrs,  $\Delta = 3$  min,  $\alpha = 1.1$ , and  $E[L] = 30$  min, 3-point derivatives). (a) Varying  $T$ . (b) Varying  $\Delta$ . (c) Varying  $\alpha$ . (d) Varying  $E[L]$ .

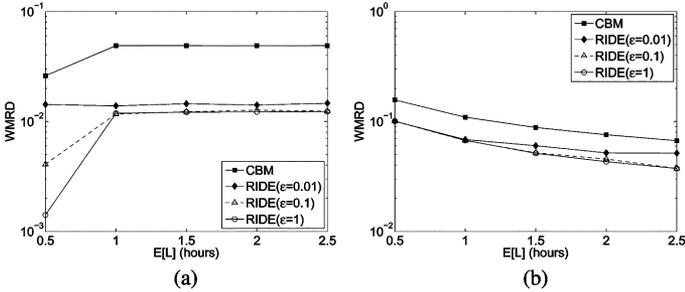


Fig. 12. Comparing CBM and RIDE for uniform and Weibull lifetimes (3-point derivatives). (a) Uniform. (b) Weibull.

### A. Models

For general P2P networks where CBM subsampling is impossible, we assume that the algorithm performs full crawls of the system every  $\Delta$  time units in the interval  $[t_0, t_0 + T]$  and then keeps probing peers that were present in the system at time  $t = t_0 + T$  until they die or their observed lifetime exceeds  $T$  time units. Based on these rules, we formulate in the next theorem the overhead for CBM.

*Theorem 8:* Total bandwidth overhead of  $(\Delta, T)$ -sampling using CBM is given by

$$b_{\text{CBM}} = \frac{C|S_0|}{\Delta} \left( T + \int_0^T [H(T) - H(x)] dx \right) \quad (43)$$

where  $|S_0|$  is the number of alive users in the system,  $C$  is the cost of probing or crawling a user, and  $H(x)$  is the CDF of residual lifetimes.

*Proof:* Denote by  $b_1, b_2$  the overhead of CBM in intervals  $[t_0, t_0 + T]$  and  $[t_0 + T, t_0 + 2T]$ , respectively. For all  $T/\Delta$  points in the first interval, each snapshot will capture  $|S_0|$  users and thus the overhead is simply  $b_1 = C|S_0|T/\Delta$ .

Now, we examine the overhead in the second interval. Consider the snapshot that captures  $|S_0|$  users at time  $t_0 + T$ , i.e.,

the last snapshot in the first interval. Denote by  $Z$  the remaining sampling duration of a random user alive at  $t_0 + T$ . Then, the expected overhead  $b_2$  incurred by probing in  $[t_0 + T, t_0 + 2T]$  is given by

$$b_2 = \frac{C|S_0|E[Z]}{\Delta}. \quad (44)$$

Next, we focus on deriving  $E[Z]$ . Denote by  $A$  the age of a user at time  $t_0 + T$ . From time  $t_0 + T$ , we only need to track peers whose age  $A$  is less than  $T$ , since  $E[Z|A \geq T] = 0$ . Thus,  $E[Z]$  can be rewritten as follows:

$$\begin{aligned} E[Z] &= \int_0^\infty E[Z|A=x] f_A(x) dx \\ &= \int_0^T E[Z|A=x] f_A(x) dx, \end{aligned} \quad (45)$$

where  $E[Z|A=x]$  is the expected sampling duration of a user given that its age is  $x < T$ . Denote by  $R(x)$  the residual lifetime of a randomly selected user given that its age  $A=x$ . Note that from time  $t_0 + T$ , we keep probing a user with age  $x < T$  until either it dies, i.e., for  $R(x)$  time units, or the remaining observation duration  $T-x$  expires, whichever happens first. Thus, it follows that

$$\begin{aligned} E[Z|A=x] &= E[\min(R(x), T-x)] \\ &= \int_0^\infty P(\min(R(x), T-x) > y) dy \\ &= \int_0^{T-x} P(R(x) > y) dy. \end{aligned} \quad (46)$$

Note that  $P(R(x) > y)$  can be expressed in terms of  $F(x)$ :

$$P(R(x) > y) = \frac{P(L > x+y)}{P(L > x)} = \frac{1-F(x+y)}{1-F(x)},$$

which together with (45), (46) gives

$$\begin{aligned} E[Z] &= \int_0^T \int_0^{T-x} \frac{1-F(x+y)}{1-F(x)} dy f_A(x) dx \\ &= E[L] \int_0^T \frac{H(T) - H(x)}{1-F(x)} f_A(x) dx. \end{aligned} \quad (47)$$

The second equality in (47) comes from the fact that  $H(x) = 1/E[L] \int_0^x (1-F(u)) du$ . Substituting  $f_A(x) = (1-F(x))/E[L]$  into (47) establishes:

$$E[Z] = \int_0^T [H(T) - H(x)] dx,$$

from which we obtain  $b_2$ . Summing up  $b_1, b_2$  gives (43). ■

Next, we examine the overhead of RIDE sampling. Note that RIDE only probes users that are captured in the first crawl until they die or  $T$  expires. Taking into account  $\epsilon$ -subsampling, we have the following theorem.

*Theorem 9:* Total bandwidth overhead of  $(\Delta, T)$ -sampling using RIDE is given by

$$b_{\text{RIDE}} = \frac{C|S_0|}{\Delta} \left( \Delta + \epsilon \int_0^T [1-H(x)] dx \right) \quad (48)$$

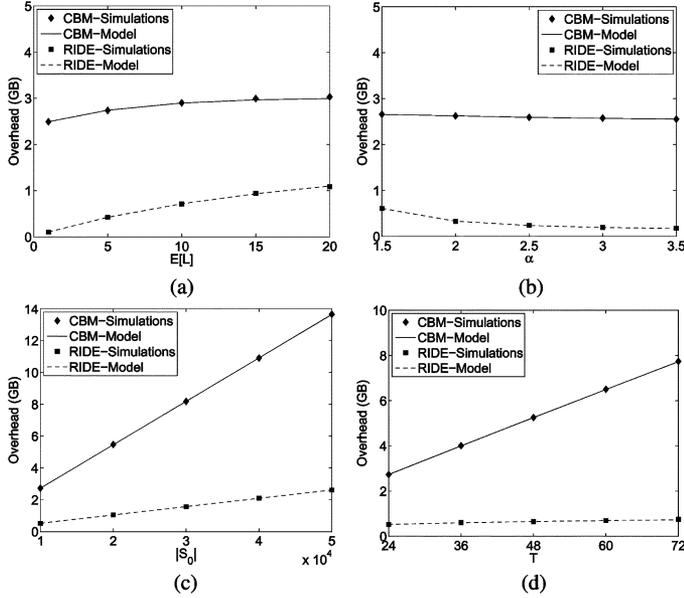


Fig. 13. Verification of models (43) and (48) against simulations. (a) Varying  $E[L]$ . (b) Varying  $\alpha$ . (c) Varying  $|S_0|$ . (d) Varying  $T$ .

where  $\epsilon$  is the fraction of peers retained in the initial set  $S_0$ .

*Proof:* In RIDE, we have one full crawl and then each user is sampled until it dies or  $T$  expires. The average duration a user survives in the system is  $E[\min(R, T)]$  given by

$$E[\min(R, T)] = \int_0^T P(\min(R, T) > x) dx \quad (49)$$

where for  $x \leq T$

$$P(\min(R, T) > x) = P(R > x) = 1 - H(x).$$

With subsampling, we have one full crawl of cost  $C|S_0|$  and then repeated crawls over systems of size  $\epsilon|S_0|$ , which gives us

$$b_{RIDE} = C|S_0| \left[ 1 + \frac{\epsilon E[\min(R, T)]}{\Delta} \right], \quad (50)$$

which together with (49) leads to (48).  $\blacksquare$

## B. Simulations and Discussion

We first verify models (43) and (48) in simulations and compare RIDE to CBM in terms of overhead. We set  $C = 1$  KB and  $\Delta = 6$  minutes in (43), (48). When simulating RIDE, we do not use subsampling and set  $\epsilon = 1$ . Users have lifetimes drawn from a power-law distribution and each departure event triggers a new arrival. Additional simulations show that this simplified treatment of departure times does not affect the result.

We plot in Fig. 13 the overhead of CBM and RIDE obtained from both simulations and models (43), (48) by fixing three parameters from the set  $\alpha = 3$ ,  $E[L] = 2$  hours,  $|S_0| = 10^4$ ,  $T = 24$  hours and varying the fourth. The figure shows that both models track simulation results pretty accurately for all studied cases and that RIDE (even without subsampling) exhibits significantly less overhead than CBM. The curves in Fig. 13(a)–(b) are almost horizontal, predicting that neither method is very sensitive to the changes in  $E[L]$  or  $\alpha$ . Fig. 13(c)–(d) displays linear increasing curves for both methods, but CBM's slope is significantly more aggressive.

TABLE I  
COMPARISON OF OVERHEAD USING  $E[L] = 1$  hour,  $\Delta = 3$  minutes

$\alpha$	$T$	$q(0.1)$	$q(0.01)$	$\alpha$	$T$	$q(0.1)$	$q(0.01)$
1.1	24 hrs	16	125	2	24 hrs	71	319
	48 hrs	17	151		48 hrs	116	573
	72 hrs	18	164		72 hrs	157	811

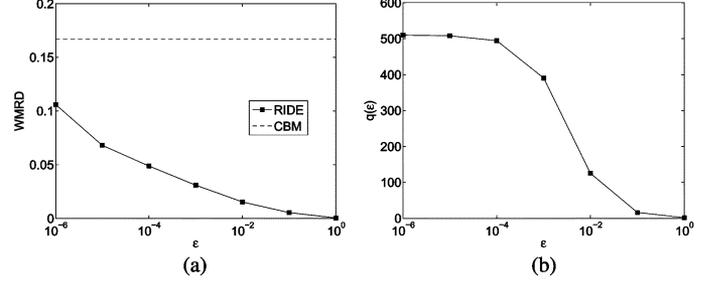


Fig. 14. Effect of  $\epsilon$  on the accuracy and overhead of RIDE subsampling in simulations (40-point derivatives). (a) WMRD. (b) Overhead ratio.

Next, define  $q(\epsilon)$  to be the ratio  $b_{CBM}/b_{RIDE}$  for the same  $|S_0|$ . Assuming Pareto lifetimes with shape  $\alpha$ , Table I shows the exact savings gained by using residual subsampling. The table shows that RIDE can reduce traffic overhead by a factor of 16–800 compared to CBM depending on the tail weight of  $F(x)$ , sampling duration  $T$ , and subsampling factor  $\epsilon$ .

As long as  $\epsilon|S_0|$  is sufficiently large, RIDE has the same accuracy as its original (non-subsampled) version, but at significantly smaller overhead. In practice, one can choose  $\epsilon$  based on the size of the initial set  $S_0$  such that  $\epsilon|S_0|$  is fixed at some pre-determined value, which can be computed using standard methods of statistical inference for any given accuracy requirement specified in terms of confidence intervals [3]. Given this dynamic selection of  $\epsilon$ , it becomes clear that RIDE can scale to arbitrarily large systems since it requires monitoring only a fixed number of users that does not depend on system size  $|S_0|$ .

We finish this section with illustrating how  $\epsilon|S_0|$  affects the accuracy and overhead of RIDE. We use different values of  $\epsilon$  in simulations with a fixed set of parameters:  $C = 1$  KB,  $\Delta = 3$  minutes,  $\alpha = 1.1$ ,  $E[L] = 1$  hour,  $|S_0| = 10^{10}$ , and  $T = 24$  hours. Fig. 14 plots the resulting metrics  $\delta$  and  $q(\epsilon)$ . The figure shows that as  $\epsilon|S_0|$  increases, RIDE's  $\delta$  tends to zero. By tuning  $\epsilon$ , the measurement application can decide how to solve the tradeoff between accuracy and overhead. For the specific example in Fig. 14, RIDE with  $\epsilon = 0.01$  reduces overhead by two orders of magnitude compared to CBM while keeping the corresponding WMRD at 10% of CBM's. These observations suggest that RIDE is indeed more suitable for large systems and long measurements than CBM.

## VII. EXPERIMENTS

In what follows in this section, we apply the residual-based algorithm to crawl the Gnutella network and estimate the distributions of peer/link lifetimes.

### A. Gnutella Crawler

Recent Gnutella networks are implemented in a two-tier structure that contains ultrapeers and leaves. Ultrapeers are responsible for forwarding search requests between each other, while leaves stay at the “edge” of the network and connect

TABLE II  
COMPARISON OF P2P MEASUREMENT STUDIES

Approach	Measured network	Interval $\Delta$	Duration $T$	Peers seen		Crawling machines	Year crawled	Connections per minute
				Probed	Crawled			
GnuSpider	Gnutella	3 min	24 – 72 hrs	–	6.4 million	1	2006	400,000
Stutzbach <i>et al.</i> [20]	Gnutella	7 min	48 hrs	–	1.3 million	7	2004	22,500
Liang <i>et al.</i> [14]	FastTrack	5 min	65 hrs	965	–	N/A	2004 – 2005	N/A
Bustamante <i>et al.</i> [2]	Gnutella	21 min	7 days	–	500,000	17	2003	< 5000
Bhagwan <i>et al.</i> [1]	Overnet	20 min	7 days	2400	–	1	2003	N/A
Chu <i>et al.</i> [5]	Gnutella	10 min	9 wks	5000	–	1	2002	N/A
Ripeanu <i>et al.</i> [16]	Gnutella	hours	–	–	48,195	N/A	2001	< 1000
Saroiu <i>et al.</i> [18]	Gnutella	7 min	60 hrs	17,125	–	N/A	2001	N/A

to several ultrapeers that provide them with search capabilities. A recent extension to the Gnutella protocol provides a crawler-friendly mechanism: upon receiving a crawl request (i.e., a handshake message with the “Crawler” field), a Gnutella client replies with a complete list of the identities of its neighboring peers.

To sample lifetimes of real Internet users using RIDE, we designed and built a scalable Gnutella crawler called GnuSpider that can operate in networks with millions of hosts and maintain reasonably small values of sampling period  $\Delta$ . As most other crawlers, GnuSpider starts the crawl using a default seed file of ultrapeers and then contact them to obtain their neighbor lists, which are then used in a BFS search to discover all currently alive ultrapeers in the system. Neighbor lists in Gnutella include other ultrapeers with whom a connection is currently active, suggested ultrapeers who may or may not be online, and leaf peers currently attached as children. The crawler records leaf peers for statistical purposes, but only contacts nodes found in the other two lists.

Our GnuSpider implementation is a single-threaded Windows process that uses asynchronous completion ports (IOCP) to manage up to 60,000 simultaneous connections to other hosts. To reduce the effect of timeouts and allow scalability, GnuSpider limits all TCP connection timeouts to 9 seconds, includes a low-overhead management of the BFS queue, and avoids socket re-binding between connections. Fig. 15(a)–(b) show bandwidth consumption in one crawling example and the number of connections per second generated by the crawler. As seen in the figure, the crawler downloads data at sustained rates of 30 mb/s and attempts on average 400,000 connections per minute. Since a certain percentage of SYN requests encounter dead or firewalled peers, the number of *successful* ultrapeer contacts lingers at 216,000/min.

Experiments with GnuSpider show that we can cover the entire Gnutella network in 3 minutes and typically discover close to 6.4 million users in the process (1.2 million of which are the ultrapeers that we attempt to contact and 5.2 million are leaf nodes). During the first 120 seconds of the crawl, the discovery rate of new leaves shown in Fig. 15(c) varies between 40,000/second and 10,000/second and that of new ultrapeers stays on average at 3000/second. It can also be seen from the figures that the last 60 seconds of the crawl usually produce a very small number of new peers since most of these connections experience a timeout. As illustrated in Fig. 15(d), 90% of ultrapeers (i.e., 1.1 million) and leaf nodes (i.e., 4.5 million) can be discovered in just 100 seconds.

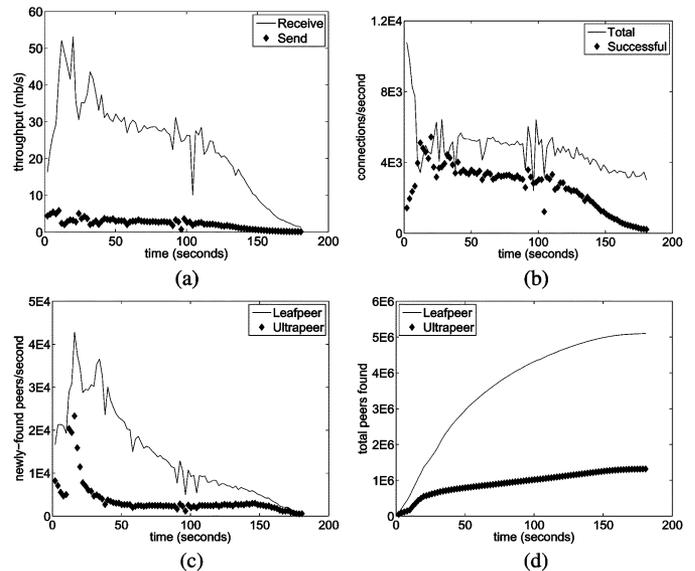


Fig. 15. Statistics of a 3-minute crawl on July 22, 2006 (single-core, dual-CPU Xeon computer @ 3 GHz). (a) Throughput. (b) Connection rate. (c) Discover rate. (d) Success percentage.

Comparison of GnuSpider to crawlers in prior experimental P2P work is shown in Table II, which provides the sampling period  $\Delta$ , window duration  $T$ , the number of peers periodically probed with SYN packets or discovered during an actual crawl, and the crawling speed in terms of contacted hosts per minute. Observe in the table that GnuSpider is not only 18 times faster than the fastest crawler in prior literature [20], but it also discovers almost 5 times more concurrent users than any other study.

### B. Peer Lifetimes

Users arriving into Gnutella immediately attempt to establish several neighboring connections to other peers currently in the system to increase their own resilience and enable themselves to route requests into the network. However, since leaves and users behind firewalls do not generally accept connection requests, selection of neighbors is often limited to non-firewalled, or as we call them *responsive*, ultrapeers.<sup>4</sup> Therefore, measurement of responsive ultrapeers provides the most useful information about the lifetime of future neighbors acquired by arriving users and allows parameter selection for existing P2P models based

<sup>4</sup>The Gnutella protocol suggest that peers behind firewalls should not become an ultrapeer. But in our measurement, about 5% of users behind firewalls act as an ultrapeer.

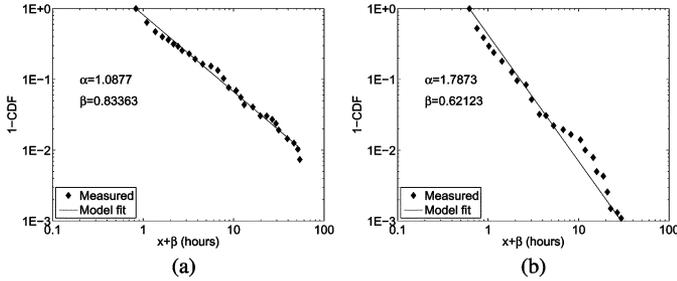


Fig. 16. Inverse-averaged estimator  $E_R(x)$  for responsive peers and links in Gnutella. Both cases use 3-point derivatives. (a) Responsive ultrapeer. (b) All links.

on lifetime distributions [12], [13], [23]. Thus, our experiments below focus only on lifetimes of ultrapeers that respond to our connection requests and the links associated with them.

To measure peer lifetimes for the plots shown below, we first obtained using GnuSpider the initial set  $S_0$  of about 468 thousand responsive ultrapeers and subsampled it using  $\epsilon = 0.213$ . Then, GnuSpider probed  $\epsilon|S_0| = 100\,000$  users for  $T = 72$  hours checking if each peer was alive every 3 minutes. It should be noted that we found that in our experiments that a certain amount of peers exhibited erratic behavior, i.e., they would respond to one request, then become silent for several subsequent requests, and eventually become responsive again. This phenomenon appeared when a peer either was too busy to reply or implemented a certain rate-limiting strategy. To filter out the effect of this behavior, we set a threshold  $u$  for how many times a peer must appear unresponsive before we declare that user dead. In the crawls below, we use  $u = 3$ .

After the observation window in GnuSpider had expired, an off-line program read the GnuSpider logs and applied RIDE to reconstruct  $F(x)$ . Fig. 16(a) shows the resulting inverse-averaged tail distribution  $1 - E_R(x)$  for the set of responsive ultrapeers. The figure matches well with a Pareto distribution with  $\alpha = 1.09$  and  $\beta = 0.83$ , where the shape parameter is very close to the 1.06 observed in [2]. Denote by  $r$  the expected residual lifetime conditioned on the fact that  $R(t)$  is within the observation window  $T$ , i.e.,  $r = E[R(t)|R(t) \leq 72]$ . Crawl results show that  $r = 10.5$  hours, but 5% of the peers in  $S_0$  leave the system in just 8 minutes.

We next proceed to compare the associated link lifetime distribution with that of peers in terms of  $\alpha$  and  $r$ .

### C. Link Lifetimes

It is straightforward to apply the residual-based algorithm to measure the link lifetime distribution in Gnutella networks. In the experiment of Section VII-B, GnuCrawler kept track of the links of responsive ultrapeers found in  $S_0$  and updated their status (i.e., connected or broken) in subsequent crawls. Using this information, we applied the same processing program to extract link residuals from GnuSpider logs and perform the proposed recovery technique to obtain  $E_R(x)$ . Fig. 16(b) shows that the resulting distribution of all link lifetimes is also power-law, but this time with  $\alpha = 1.79$ , which is much larger than that in the peer lifetime distribution. This observation establishes that *the lifetime of a link is probabilistically smaller than that of a*

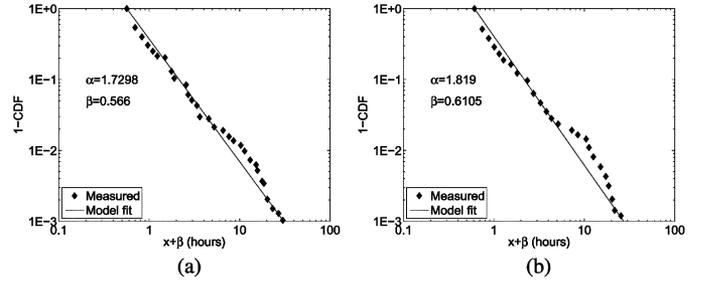


Fig. 17. Inverse-averaged estimator  $E_R(x)$  for different types of links in Gnutella. Both cases use 3-point derivatives. (a) Ultra-leaf links. (b) Ultra-ultra links.

*peer and one may expect more frequent changes in neighboring relationships.* We also find that  $r$  is 3.8 hours and 16.4% of links disappear within 8 minutes.

Next, we treat the links between ultrapeers and leaves separately from those among ultrapeers and plot the corresponding distributions in Fig. 17. Interestingly, the figure shows that the ultra-leaf links are slightly more stable (i.e., exhibit a heavier tail) than ultra-ultra links: the former has  $\alpha = 1.73$  and the latter has  $\alpha = 1.82$ ; the conditional expected lifetimes  $r$  of the two types of links are 3.9 and 3.5 hours, respectively. This can be plausibly explained by the fact that a leaf is usually inactive in collecting information about alternate ultrapeers and is thus less likely to switch its attachment point.

### D. Discussion

With the experimental results of this section, we are now able to study resilience properties of Gnutella networks by applying models from [13], which use the average residual link lifetime and average node degree  $d$  as input parameters. Given  $d = 28.5$  neighbors observed in our experiments and a 1-minute failed-neighbor replacement delay, we obtain that the probability for the network to disconnect at the ultrapeer level is below  $10^{-64}$ . However, leaves may be isolated with a non-negligible probability, because they only have one or two attachment points, i.e.,  $d \leq 2$ , which we plan to explicitly study in future work.

## VIII. RELATED WORK

Some of the first P2P sampling studies date to 2001 [16], [18] and the first use of CBM can be traced to Saroiu *et al.* [18] who sampled 17 000 Gnutella peers every 7 minutes using TCP SYN packets over a period of 60 hours. In a follow-up effort in [5], Chu *et al.* used a similar method, but probed 5000 peers every 7 minutes for 10 weeks. Bhagwan *et al.* [1] improved over [5], [18] by implementing the Overnet protocol and probing a randomly chosen subset of peers in the system to measure their availability (i.e., the portion of time they were present online). Their experiment selected 2400 out of around 90 000 peers and kept probing them every 20 minutes for 7 days. Liang *et al.* [14] measured lifetime distributions of links in the KaZaa network, but these experiments were limited to the connections passing through the authors' monitoring hosts.

More related work can be found in [2] and [20]. Bustamante *et al.* [2] implemented a Gnutella sampler using 17 monitoring clients that periodically probed 500 000 peers in the network every 21 minutes for 7 days. In more recent work, Stutzbach *et al.*

*al.* [20] developed a much faster crawler that in 2004 was able to cover the entire Gnutella network of 158 000 ultrapeers within 7 minutes. The closest approach to understanding sampling bias is another recent paper by Stutzbach *et al.* [21], which focused on capturing unbiased snapshots of joint properties of users currently alive in P2P systems using random walks.

## IX. CONCLUSION

In this paper, we showed that direct lifetime sampling suffered from estimation bias and did not admit any fundamental improvement besides reducing probing interval  $\Delta$ . To overcome this limitation, we proposed and analyzed a novel residual-based lifetime sampling algorithm, which measured lifetime distributions with high accuracy and required several orders of magnitude less bandwidth than the prior approaches. Using this method, we sampled Gnutella users and discovered that lifetimes of peers and links exhibited power-law distributions, but with different shape parameters, where links are indeed much more volatile than actual peers.

## REFERENCES

- [1] R. Bhagwan, S. Savage, and G. M. Voelker, "Understanding availability," in *Proc. IPTPS*, Feb. 2003, pp. 256–267.
- [2] F. E. Bustamante and Y. Qiao, "Friendships that last: Peer lifespan and its role in P2P protocols," in *Proc. Int. Workshop on Web Content Caching and Distribution*, Sep. 2003.
- [3] G. Casella and R. L. Berger, *Statistical Inference*, 2nd ed. New York: Duxbury/Thomson Learning, 2002.
- [4] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker, "Making Gnutella-like P2P systems scalable," in *Proc. ACM SIGCOMM*, Aug. 2003, pp. 407–418.
- [5] J. Chu, K. Labonte, and B. N. Levine, "Availability and locality measurements of peer-to-peer file systems," in *Proc. ITCOM Conf.*, Jul. 2002, vol. 4868, pp. 310–321.
- [6] N. Duffield, C. Lund, and M. Thorup, "Estimating flow distributions from sampled flow statistics," in *Proc. ACM SIGCOMM*, Aug. 2003, pp. 325–336.
- [7] R. J. Dunn, J. Zahorjan, S. D. Gribble, and H. M. Levy, "Presence-based availability and P2P systems," in *Proc. IEEE Int. Conf. Peer-to-Peer Computing*, Aug. 2005, pp. 209–216.
- [8] Z. Ge, D. R. Figueiredo, S. Jaiswal, J. Kurose, and D. Towsley, "Modeling peer-peer file sharing systems," in *Proc. IEEE INFOCOM*, Mar. 2003, vol. 3, pp. 2188–2198.
- [9] Gnutella. [Online]. Available: <http://www.gnutella.com/>
- [10] K. Gummadi, R. Gummadi, S. Gribble, S. Ratnasamy, S. Shenker, and I. Stoica, "The impact of DHT routing geometry on resilience and proximity," in *Proc. ACM SIGCOMM*, Aug. 2003, pp. 381–394.
- [11] KaZaA. [Online]. Available: <http://www.kazaa.com/>
- [12] S. Krishnamurthy, S. El-Ansary, E. Aurell, and S. Haridi, "A statistical theory of chord under churn," in *Proc. IPTPS*, Feb. 2005, pp. 93–103.
- [13] D. Leonard, V. Rai, and D. Loguinov, "On lifetime-based node failure and stochastic resilience of decentralized peer-to-peer networks," in *Proc. ACM SIGMETRICS*, Jun. 2005, pp. 26–37.
- [14] J. Liang, R. Kumar, and K. W. Ross, "The FastTrack overlay: A measurement study," *Computer Networks*, vol. 50, no. 6, pp. 842–858, Apr. 2006.
- [15] S. Resnick, *Adventures in Stochastic Processes*. New York: Birkhäuser, 2002.

- [16] M. Ripeanu, I. Foster, and A. Iamnitchi, "Mapping the Gnutella network: Properties of large-scale peer-to-peer systems and implications for system design," *IEEE Internet Comput. J.*, vol. 6, no. 1, pp. 50–57, Jan.–Feb. 2002.
- [17] 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.
- [18] S. Saroiu, P. K. Gummadi, and S. D. Gribble, "A measurement study of peer-to-peer file sharing systems," in *Proc. SPIE/ACM Multimedia Computing and Networking*, Jan. 2002, vol. 4673, pp. 156–170.
- [19] S. Sen and J. Wang, "Analyzing peer-to-peer traffic across large networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 2, pp. 219–232, Apr. 2004.
- [20] D. Stutzbach and R. Rejaie, "Understanding churn in peer-to-peer networks," in *Proc. ACM IMC*, Oct. 2006, pp. 189–202.
- [21] D. Stutzbach, R. Rejaie, N. Duffield, S. Sen, and W. Willinger, "On unbiased sampling for unstructured peer-to-peer networks," in *Proc. ACM IMC*, Apr. 2006, pp. 27–40.
- [22] D. Stutzbach, R. Rejaie, and S. Sen, "Characterizing unstructured overlay topologies in modern file-sharing systems," in *Proc. ACM IMC*, Oct. 2005, pp. 49–62.
- [23] Z. Yao, D. Leonard, X. Wang, and D. Loguinov, "Modeling heterogeneous user churn and local resilience of unstructured P2P networks," in *Proc. IEEE ICNP*, Nov. 2006, pp. 32–41.

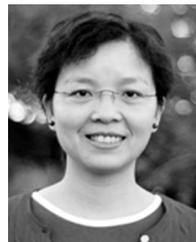


**Xiaoming Wang** (S'04) received the B.S. degree in computer science and the M.S. degree in electronic engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 1999 and 2002, respectively. He is currently working toward the Ph.D. degree at Texas A&M University, College Station.

During 2002–2003, he worked for Samsung Advanced Institute of Technology, South Korea. His research interests include peer-to-peer systems, probabilistic analysis of computer networks, and topology

modeling.

Mr. Wang is a student member of the ACM.



**Zhongmei Yao** (S'06) received the B.S. degree in engineering from Donghua University (formerly China Textile University), Shanghai, China, in 1997 and the M.S. degree in computer science from Louisiana Tech University, Ruston, in 2004. She is currently pursuing the Ph.D. degree in the Department of Computer Science at Texas A&M University, College Station.

Her research interests include peer-to-peer systems, Markov chains, and stochastic network modeling.

Ms. Yao is a student member of the ACM.



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

Between 2002 and 2007, he was an Assistant Professor in the Department of Computer Science at Texas A&M University, College Station. He is currently a tenured Associate Professor and Director of the Internet Research Lab (IRL) in the same department. His research interests include peer-to-peer

networks, video streaming, congestion control, Internet measurement and modeling.

Dr. Loguinov is a senior member of the ACM.