Robust Lifetime Measurement in Large-Scale P2P Systems with Non-Stationary Arrivals

Xiaoming Wang

Joint work with Zhongmei Yao, Yueping Zhang, and Dmitri Loguinov

Internet Research Lab
Computer Science and Engineering Department
Texas A&M UniversityMay

Sept. 10, 2009

Agenda

- Introduction
  - Background and motivation
- Previous approaches
  - CBM and RIDE
- Proposal of new method
  - U-RIDE
- Experimental Evaluation
- Conclusion

Introduction

- Peer-to-peer networks are popular platforms for many Internet applications
  - Characterizing these systems is important for theoretic modeling of resilience, throughput, etc.
- However, many existing P2P systems are fully distributed, large-scale, and highly dynamic
- Therefore, measuring these systems is challenging
  - Limit of bandwidth and lack of infrastructural support
- The goal of this work is to address one of such challenging tasks - measuring the distribution of user lifetimes

Measuring Lifetime Distribution

- An instance of the lifetime is the duration of a user's appearance in the system
- Let L be the lifetime of a random user
  - Define $F_L(x) = P(L \le x)$ to be the CDF of the lifetime

  Our target metric

- One straightforward solution is to collect lifetime instances by periodically probing users
  - And then compute empirical distribution $E(x)$ to estimate $F_L(x)$
- Due to hardware constraint and security concern, we cannot probe all users with infinitely small intervals

Measuring Lifetime Distribution 2

- In large-scale distributed systems, it is non-trivial to measure the exact lengths of lifetime instances
  - We need a definition for accuracy
- Let Δ be the probing interval and define discrete point $x_i = i\Delta$
- Estimator $E(x)$ is unbiased if it can correctly reproduce the distribution of lifetime L at all discrete points $\{x_i\}$ for any $\Delta > 0$:
  - Our target for accuracy: $E(x_i) = F_L(x_i)$
- Probing traffic could be significant for large systems
  - Our target for overhead: small amount of probing traffic

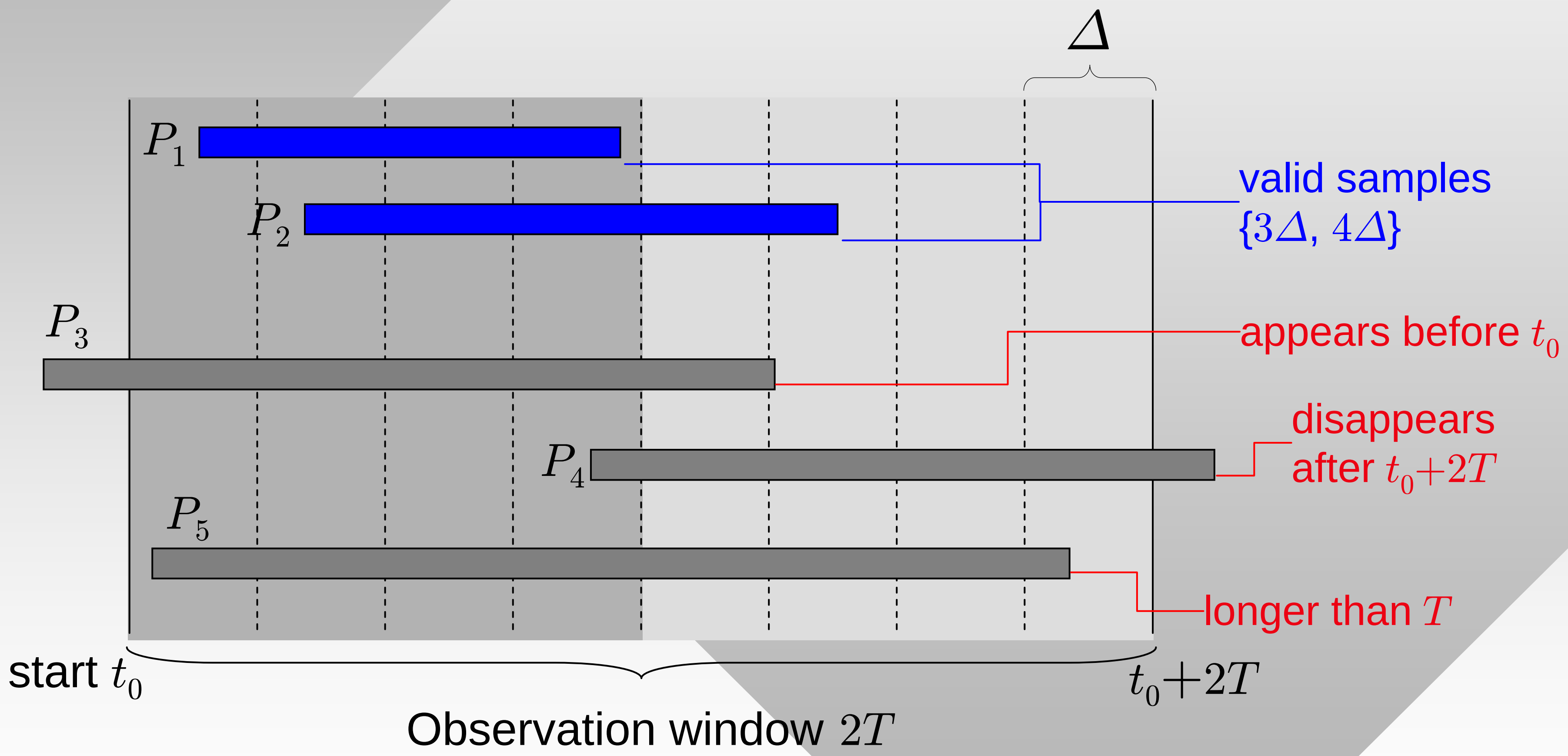
Agenda

- Introduction
  - Background and motivation
- Previous approaches
  - CBM and RIDE
- Proposal of new method
  - U-RIDE
- Experimental Evaluation
- Conclusion

Previous Methods – CBM

- Create-Based Method (CBM) uses an observation window of $2T$
  - Within the window, it takes a snapshot of the system every Δ time units
- To avoid sampling bias, CBM divides the window into two halves and only includes lifetime samples that satisfy the following conditions
  - Appear during the first half of the window
  - Disappear within the window
  - Have a lifetime no longer than T

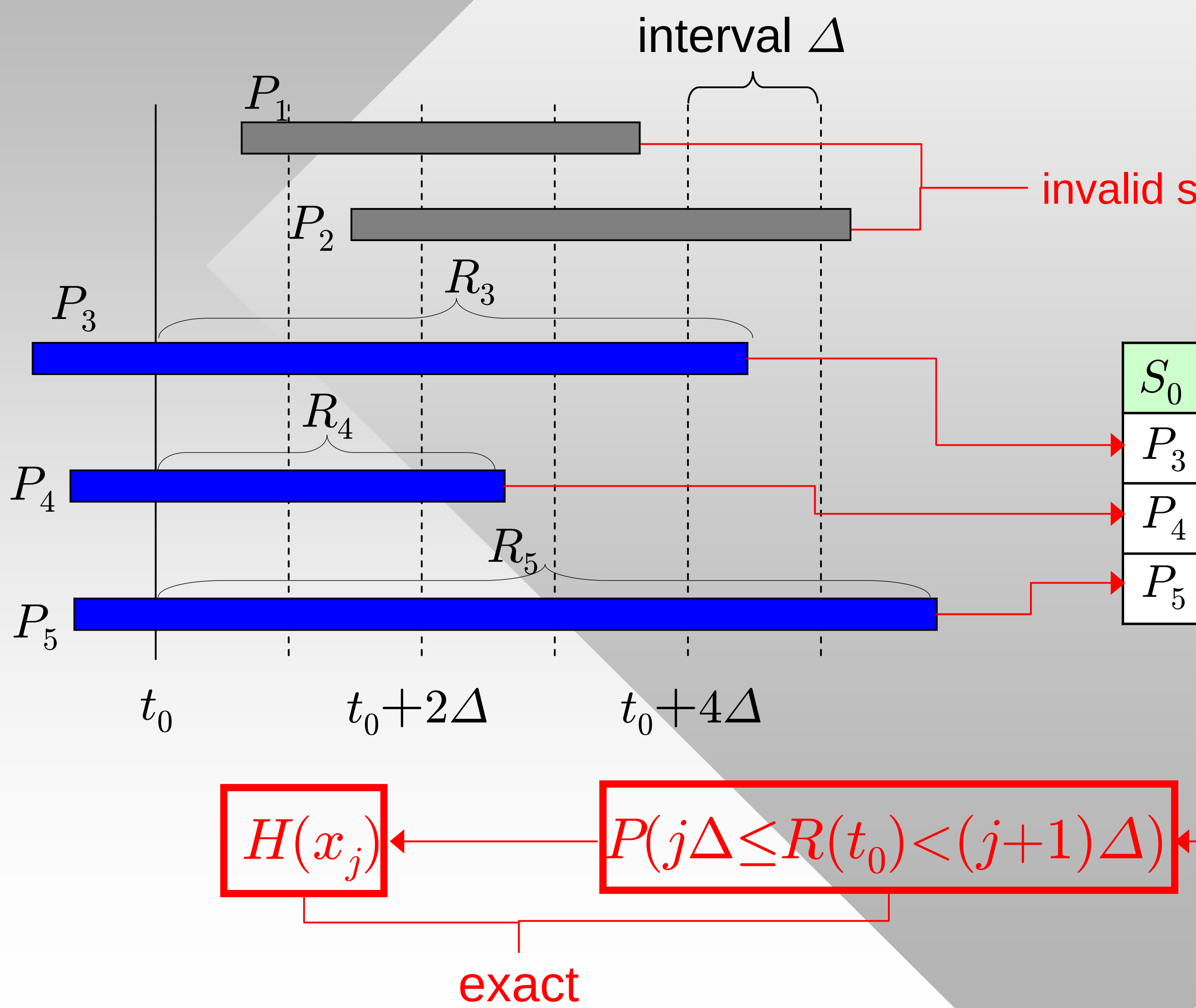
Example of CBM Sampling



Previous Methods – RIDE

- Wang *et. al*. [INFOCOM'07] proved that CBM can be arbitrarily biased in estimating the lifetime distribution
- ResIDual-based Estimator (RIDE) was proposed to address the issue in CBM
  - Take one snapshot of the system at time t_0 and record alive users in S_0
  - Probe these alive users to obtain their residuals and compute distribution $H(x)$
- Wang *et. al.* also proved that under stationary systems, RIDE can produce an unbiased estimator
  - Moreover, RIDE can be configured to incur much less traffic overhead than CBM

Example of RIDE Sampling



S_0	$R(t_0)$
P_3	$[4\Delta, 5\Delta)$
P_4	$[2\Delta, 3\Delta)$
P_5	$[5\Delta, 6\Delta)$

Agenda

- Introduction
  - Background and motivation
- Previous approaches
  - CBM and RIDE
- Proposal of new method
  - U-RIDE
- Experimental Evaluation
- Conclusion

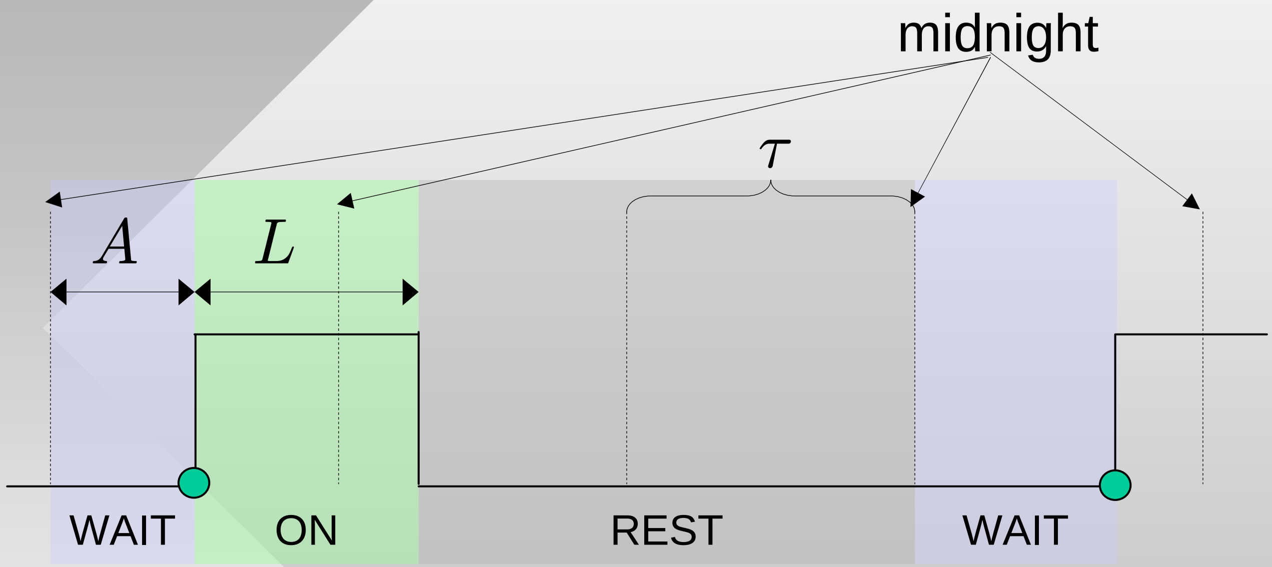
Motivation

- While RIDE can achieve both high accuracy and low overhead
  - RIDE relies on one critical assumption – stationarity of the user arrival process
- However, many systems exhibit diurnal arrival/departure patterns or any other non-stationary dynamics
  - Therefore, we need to investigate whether RIDE can achieve the same benefit mentioned earlier
- To do so, we first propose a model for non-stationary arrivals

Non-Stationary User Churn

- Our proposed model models each user with an alternating process
  - ON (online) and OFF (offline) states
- Moreover, the time is partitioned into bins of size τ
  - For example, a day is a bin
- OFF states are split into two sub-states
  - REST: the delay between the user's departure and midnight of the day when he/she joins the system again
  - WAIT: the delay from midnight until the user's arrival into the system within a given day
- Non-Stationary-Periodic Churn Model (NS-PCM)

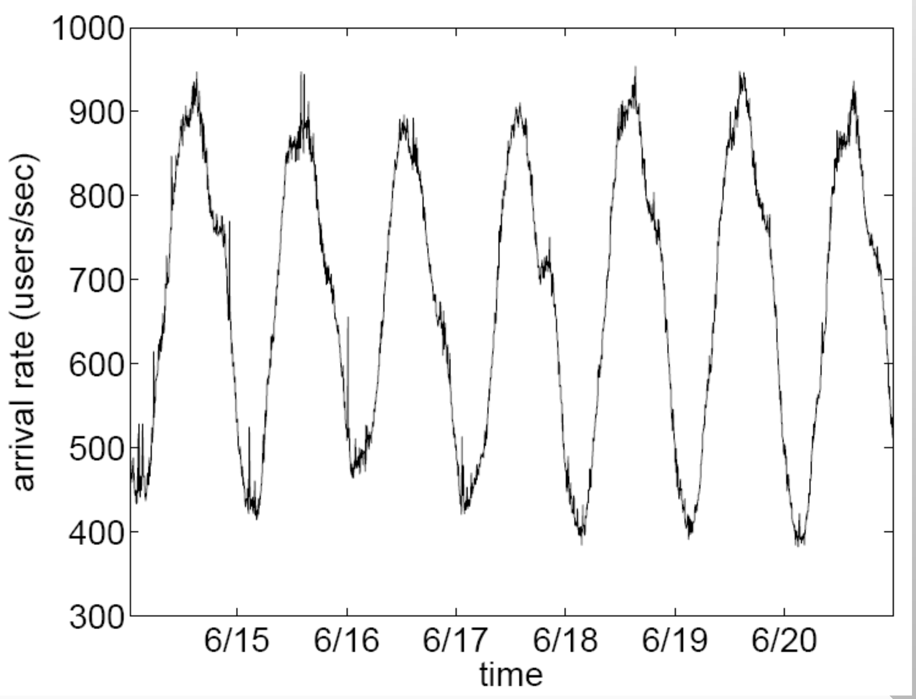
Example of User Process



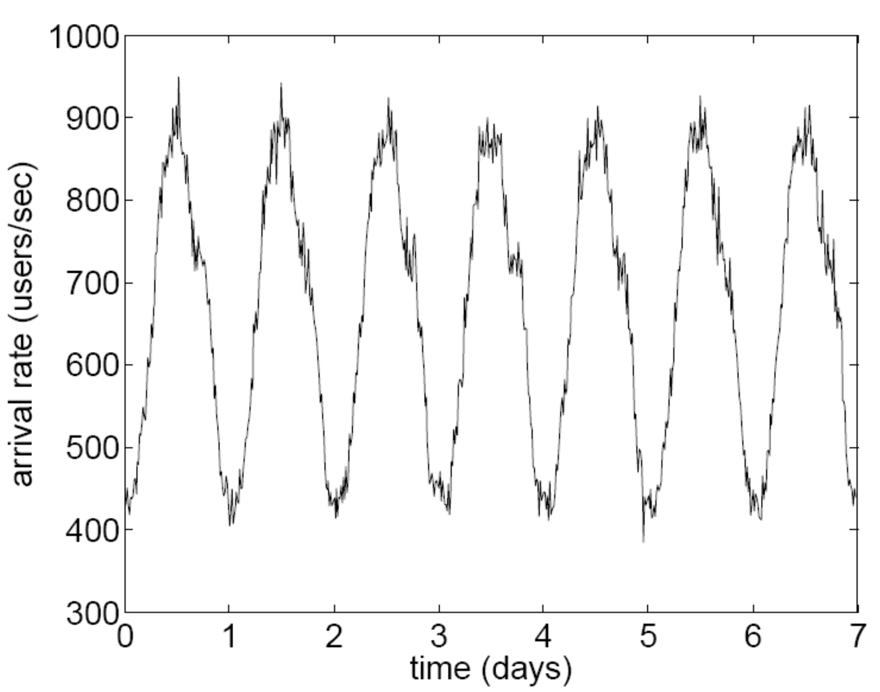
- For any time t, define $t^\star = t - \tau\lfloor t/\tau \rfloor$ to be the bin offset (remainder) of t
- For x in $[0, \tau]$, define $F_A(x) = P(A \le x)$ to be the CDF of the arrival time A

Gnutella and NS-PCM

- NS-PCM mimics the arrival process of Gnutella



Actual data measured from Gnutella



Simulations generated by NS-PCM

Analysis of RIDE in NS-PCM

- Theorem 1: Under NS-PCM, residual lifetime distribution $H(x,t_0)$ is a periodic function

$$H(x,t_0) = 1 - \frac{\int_x^\infty \omega(z-x, t_0^\star) dF_L(z)}{\int_0^\infty \omega(z, t_0^\star) dF_L(z)}$$

  - where

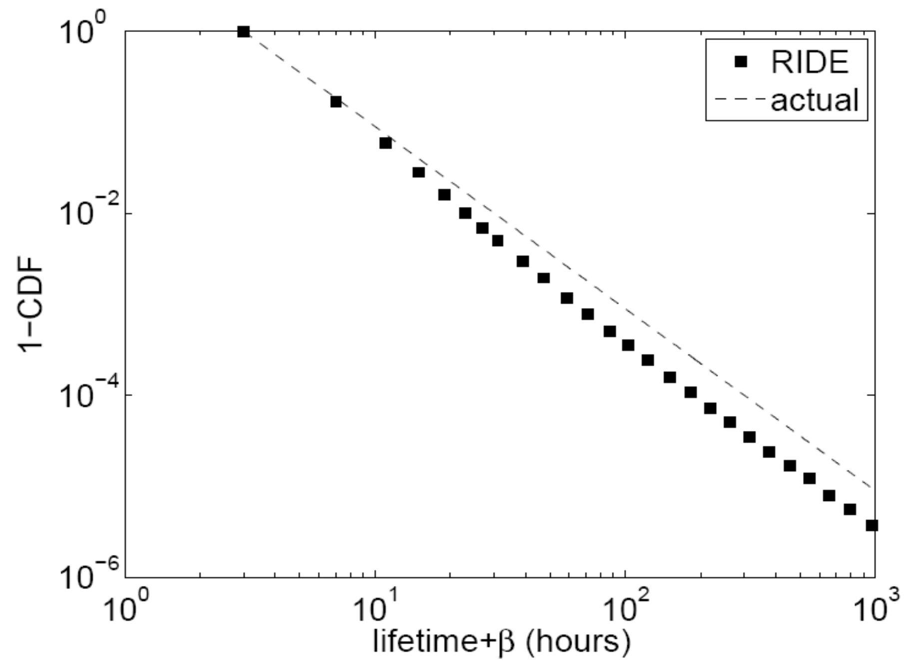
$$\omega(x,u) = F_A(u) - F_A(\max(u-x^\star, 0)) + 1 - F_A(1+\min(u-x^\star,0)) + \lfloor x/\tau \rfloor$$

- Recall in a stationary model, RIDE depends on

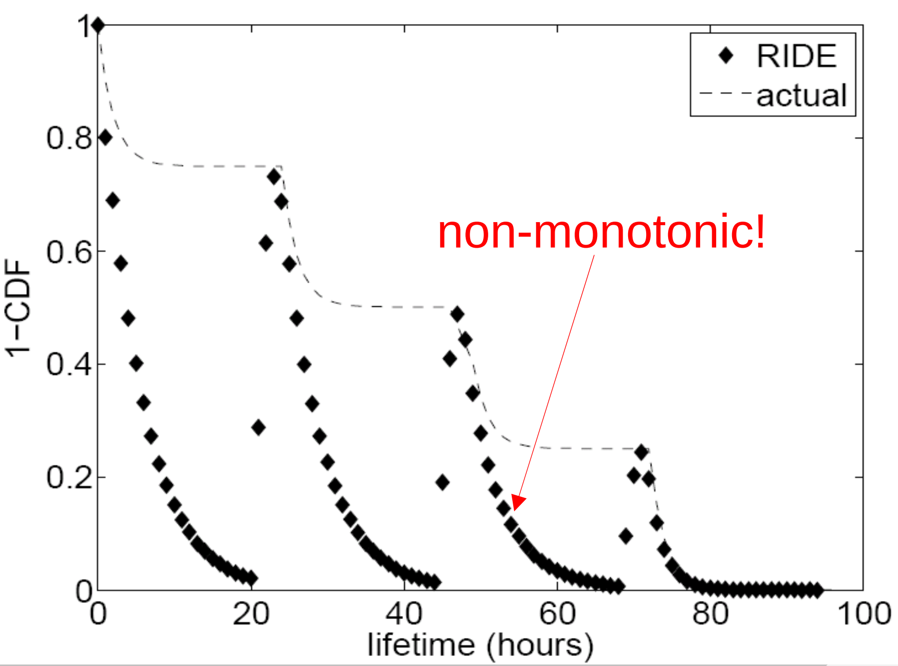
$$H(x) = \frac{1}{E[L]} \int_0^x (1 - F_L(y)) dy$$

- Therefore, RIDE is biased in NS-PCM
  - Differentiating $H(x,t_0)$ gives no simple formula of $F_L(x)$

RIDE under NS-PCM



Power-law



Periodic

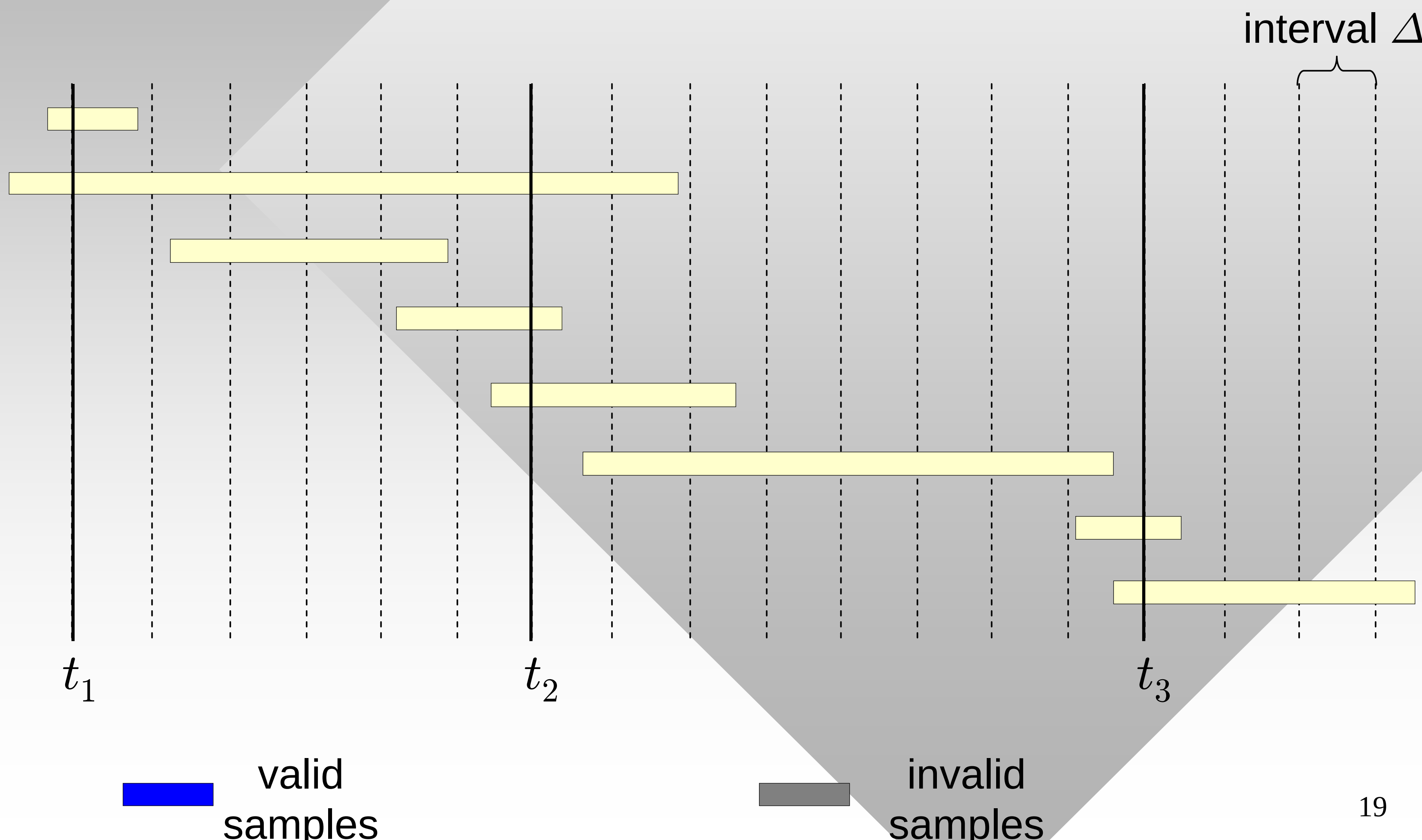
- Simulations indicate that RIDE deviates from $F_L(x)$
  - Its estimation does not even represent a valid CDF function

Proposed Sampling Algorithm - U-RIDE

- Instead of just one snapshot at time t_0, we crawl the system at multiple time points t_m $(m=1 \ldots M)$
  - Sampling schedule $T_M = \{t_1, t_2, \ldots, t_M\}$
  - Offset schedule $O_M = \{t_1^\star, t_2^\star, \ldots, t_M^\star\}$
- For the m-th snapshot, U-RIDE keeps track of captured alive users
  - $N_R(t_m)$: the number of alive users in this snapshot
  - $N_R(x, t_m)$: the number of them with residual $\leq x$
- Then, compute the ratio for each x_j

$$r(M, x_j) = \frac{\sum_{m=1}^{M} N_R(x_j, t_m)}{\sum_{m=1}^{M} N_R(t_m)}$$

Example of U-RIDE Sampling



Proposed Sampling Algorithm - U-RIDE

- We call a schedule uniform if offset schedule O_M forms a uniform distribution in $[0, \tau]$

- Theorem 2: Under a uniform schedule, the ratio r

$$E_H^*(x_j) := \lim_{M\to\infty} r(M, x_j) = \frac{1}{E[L]} \int_0^{x_j} (1 - F_L(t))dt$$

- An unbiased estimator is given by

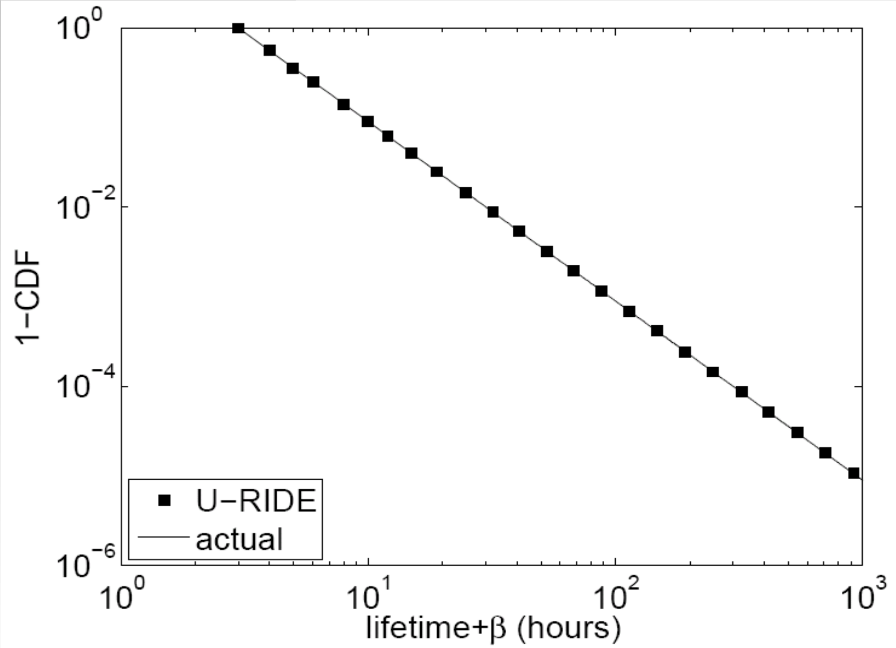
$$E_R^*(x_j) = 1 - \frac{h^*(x_j)}{h^*(0)}$$

  - where $h^*(x)$ is the numerical derivative of ratio function $E_H{}^*(x)$

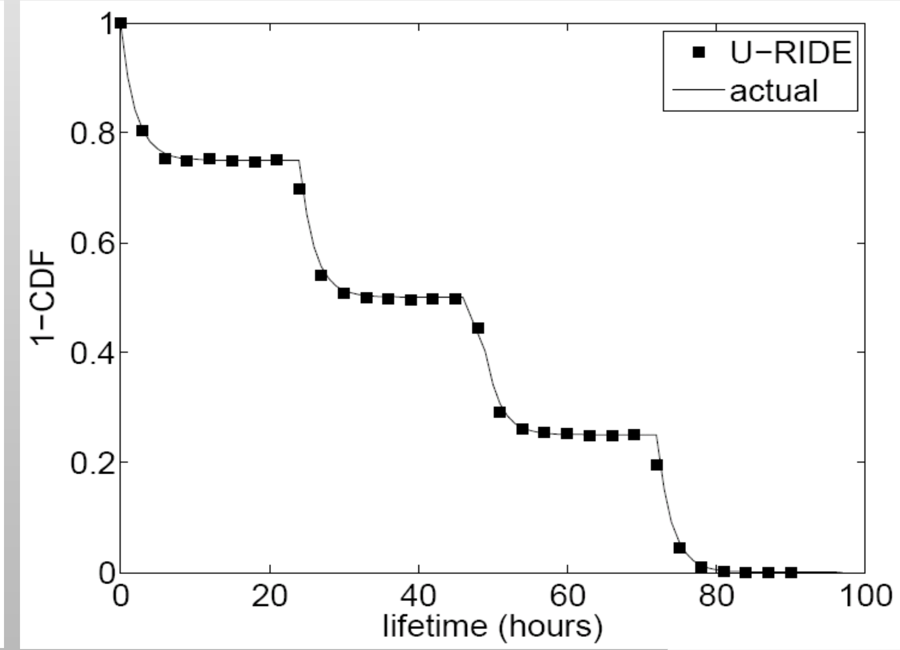
Computer Science, Texas A&M University

U-RIDE under NS-PCM

- Simulations show an exact match between U-RIDE estimation and $F_L(x)$



Power-law



Periodic

Overhead and Subsampling

- Residual sampling supports ϵ-subsampling – uniformly select ϵ percent of valid samples
  - We also prove that CBM does not support subsampling
- Wang *et. al.* [INFOCOM'07] have proved that with ϵ-subsampling, RIDE can reduce overhead by a factor of over 100 compared to CBM
- The question is whether U-RIDE can save the same amount of bandwidth

Overhead and Subsampling 2

- Theorem 3: Overhead ratio of U-RIDE and RIDE q_{UR} is

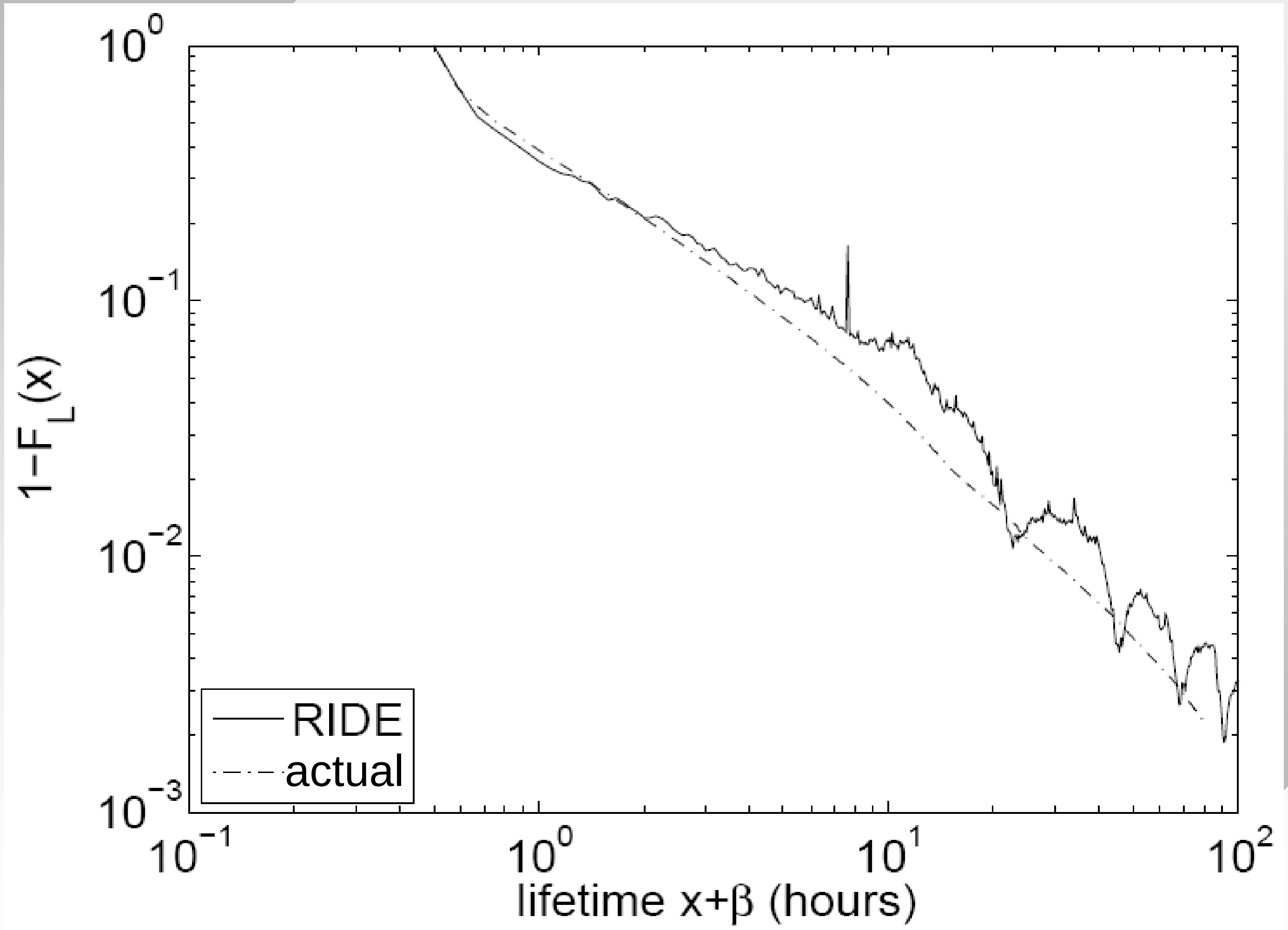
$$q_{UR} = 1 + \frac{\tau}{E[L]} \sum_{m=2}^{M} \int_{y_{m-1}}^{y_m} f_A(t^\star)(1 - F_L(y_m - t))dt$$

  - where $y_m = m\Delta(1-p)/p$ and p is a scheduling parameter
- This result shows that U-RIDE incurs more traffic overhead than RIDE in the original form
  - However, by using a smaller ϵ, U-RIDE's overhead can always be upper bounded within that of RIDE
- In fact, we can choose proper ϵ based on the size of the initial set S_0 so that $\epsilon|S_0|$ is fixed at some pre-determined value

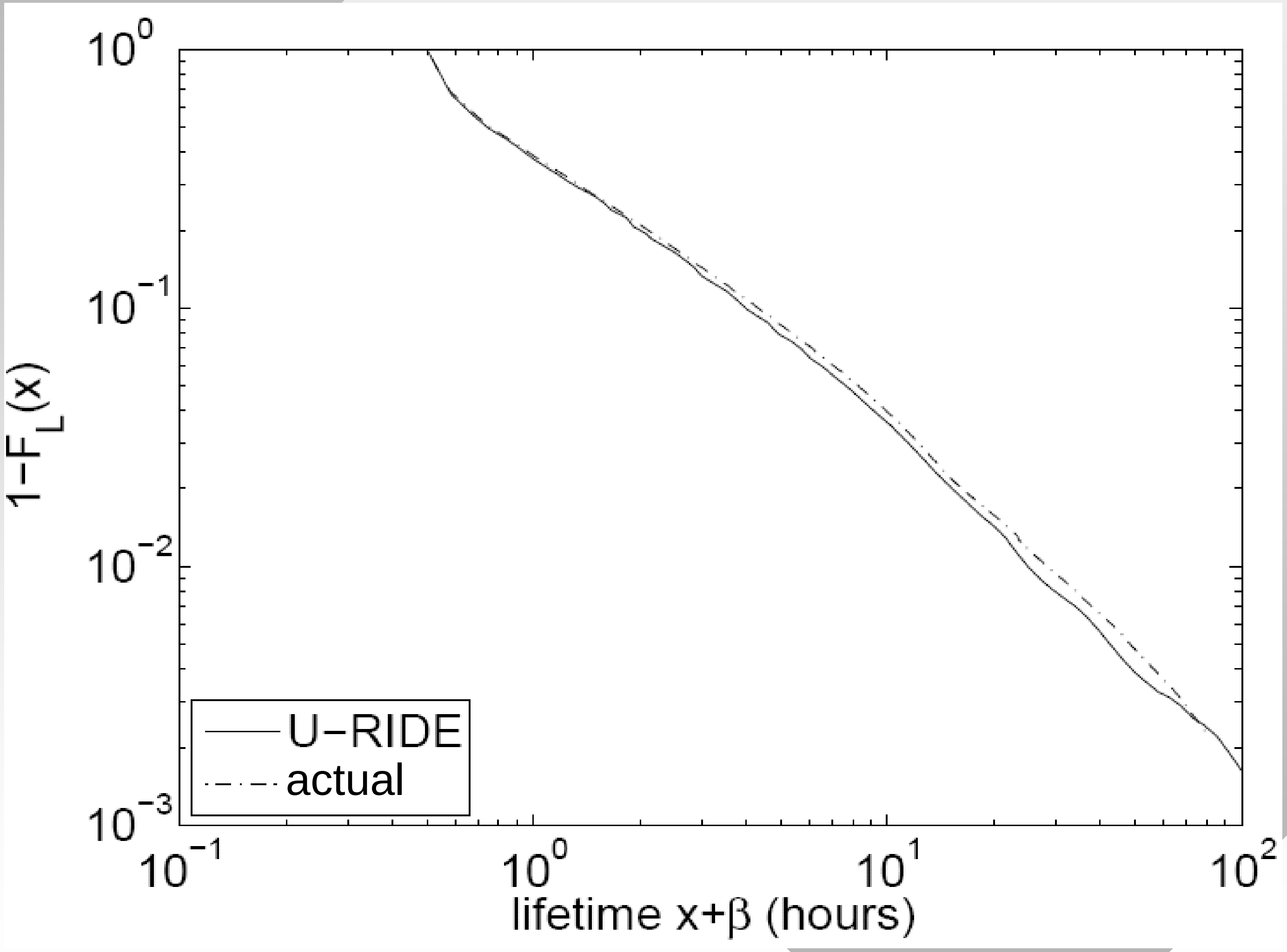
Agenda

- Introduction
  - Background and motivation
- Previous approaches
  - CBM and RIDE
- Proposal of new method
  - U-RIDE
- Experimental Evaluation
- Conclusion

Gnutella Measurement – RIDE



Gnutella Measurement – U-RIDE



Conclusion

- We studied the tradeoff between accuracy and scalability in P2P systems with non-stationary arrivals
  - We proposed a novel non-stationary churn model NS-PCM
  - We introduced a simple algorithm U-RIDE that can achieve both accuracy and scalability
- Future work includes
  - Applying NS-PCM to understand how it affects existing results in P2P
  - Extending U-RIDE for measuring the arrival process of P2P systems

The End

- Thank you!