Towards a Better Understanding of Churn in Peer-to-Peer Networks

Daniel Stutzbach, Reza Rejaie
{*agthorr, reza*}*@cs.uoregon.edu*

Abstract— **The dynamics of peer participation, or *churn*, are an inherent property of Peer-to-Peer (P2P) systems that should be incorporated in both the design and evaluation of P2P systems. This requires a proper characterization of churn in real-world P2P networks. However, the few previous measurement-based studies on the characterization of P2P systems have used either unrepresentative group of peers or coarse-grain measurements. In this extended abstract, we characterize churn in the Gnutella network based on fine-grained monitoring of the entire population. We developed a new crawler that can capture a complete snapshot of Gnutella network within a few minutes. This not only improves the accuracy, by reducing distortion in captured snapshots, but it also increase the granularity of captured dynamics. We present our preliminary characterizations of peer uptime and discuss their implications. In particular, we show that peer uptime follows a power-law distribution rather than the commonly assumed Poisson distribution. In a nutshell, a large portion of up peers are highly stable, yet the remaining peers turnover very quickly.**

Keywords—**Peer-to-Peer, Churn, Characterization, Gnutella**

I. INTRODUCTION

The Internet has witnessed a rapid increase in the popularity of Peer-to-Peer (P2P) systems during recent years. These systems use a multi-node communication model that is different from the traditional client-server model. More importantly, there is often no direct control, nor any coordination or constraint, on the participation of individual peers. Each peer joins and leaves the system at any arbitrary time. This means that the dynamics of peer participation, which are also called membership dynamics or *churn*, are an inherent property of P2P systems. Churn significantly affects both the design and evaluation of P2P systems. On the evaluation side, any P2P application should be tested under a realistic degree of membership dynamics to ensure that it properly performs in practice. On the design side, it is essential that P2P applications incorporate an adequate degree of robustness against the expected membership dynamics in order to function correctly and efficiently. A newly arrived peer should receive a proper portion of the state, and the maintained state held by a suddenly departed peer should be available from other participating peers. The basic approach to ensure robustness to churn is to add a level of redundancy by replicating the maintained state (such as part of the name space in structured P2P systems, or pointers to other peers in unstructured P2P networks) throughout the system. However, the required degree of redundancy and implementation details are different between structured and unstructured P2P systems. For example, unstructured P2P networks are known to be more resilient to churn whereas Distributed Hash Tables (DHTs) can not gracefully handle churn [1] and require some extra mechanisms [2].

To incorporate churn in the design and evaluation of P2P systems, one needs to have a "representative" characterization of churn, otherwise the incorporated degree of redundancy will either be inadequate or excessive. The former case affects the correctness of P2P applications whereas the latter case could degrade their performance. To derive a representative characterization of churn, several issues should be addressed as follows: First, such characterization should be obtained from accurate measurement of a large scale P2P system in a realistic setting such as a large group of active, heterogeneous, and geographically distributed clients such as Gnutella. Second, such measurements should capture "user-controlled" (rather than protocol-controlled) aspects of membership dynamics because this will generalize to other applications. In contrast, protocol-controlled dimensions of membership dynamics (such as conditions to join or leave, or the minimum time to download a specific type of content) are application specific, and thus not relevant to all applications. Third, it is important to characterize churn with a sufficient degree of detail so that membership dynamics can be regenerated for evaluating P2P solutions through simulation. Specifically, at a minimum this requires knowing the uptime and inter-arrival distributions.

Unfortunately, performing such characterizations for large scale P2P networks is inherently difficult. A common approach in such a measurement study is to use a crawler to capture snapshots of a P2P system. Then, comparing back-to-back snapshots can reveal changes in the system. Previous studies have either used slow crawlers (*e.g.,* one hour per crawl) or captured partial snapshot of P2P systems or used passive measurements. The resulting characterization in these approaches have limited accuracy for different reasons as follow: First, using slow crawlers introduces two limitations, *(i)* since the membership changes during a crawl, the longer a crawl takes, the more distorted the captured snapshot becomes, and *(ii)* the granularity of captured dynamics is limited by the length of each crawl. The longer a crawl takes, the larger the gap between consecutive snapshots, and thus the coarser the timescale of captured dynamics. Second, using partial snapshot of P2P systems (*i.e.,* monitoring the uptime of sampled peers) [3], [4] to characterize churn in based on the implicit assumption that peer uptime follows a Poisson distribution (*i.e.,* it is memoryless). As our results show, this assumption is not held since the majority of peers that are up are unusually long-lived peers. Therefore, this characterization of churn based in this partial can easily bias the results towards short or long lived peers depending on the sampling strategy. Third, studies that used passive measurement [5], [6], [7], can roughly approximate uptime for a subset of peers in the system that are visible from the measurement point. Therefore, their captured snapshots may not be representative and their measured uptimes depend on the typical size of transferred content among peers. Despite the absence of any reliable characterization of churn, P2P application designers often simulate P2P membership dynamics by assuming that peer uptime follows a Poisson distribution [8], [9], [10], [2].

In this extended abstract, we present our preliminary characterization of churn based on recent measurements of the Gnutella network. We focus on the Gnutella network because it is a large scale P2P network with hundreds of thousands of heterogeneous and geographically distributed peers. More importantly, since each Gnutella client is directly controlled by a user (*i.e.,* it does not run as a daemon), its behavior presents "user-driven" aspects of membership dynamics. We have devised a set of crawling techniques into a new crawler, called *Cruiser*, that enables us to capture an accurate snapshot of the Gnutella population in around 4 minutes. Since Cruiser is several orders of magnitude faster than previously reported crawlers, its captured snapshots are significantly more accurate, and the granularity of captured dynamics is much finer. Furthermore, we measure uptime in an unbiased fashion to ensure that our characterization is not skewed towards long or short lived peers. Finally, we present our preliminary characterization of peer uptime among Gnutella clients and discuss their implications. *The main contribution of this paper is to present the first detailed characterization of peer uptime based on accurate and complete snapshots of a large scale P2P network. More importantly, we show that peer uptime follows a two-piece power law distribution rather than the commonly used Poisson distribution.*

The rest of this extended abstract is organized as follows: Section II presents an overview of our measurement methodology as well as our approach to measure peer uptime in an unbiased fashion. Our preliminary characterizations of churn and their implications are presented in Section III. We briefly present related work in Section IV.

II. Measurement Methodology

Our basic measurement strategy is to perform back-to-back crawls of the Gnutella network. Given our back-to-back snapshots, we can estimate arrival and departure times for individual peers, and thus measure their uptime. The gap between the start of consecutive crawls determines the granularity of estimated peer uptimes. Therefore, our goal is to reduce duration of each crawl to improve accuracy of our measured peer uptimes. In the remaining portion of this section, we briefly describe an overview of our crawler, and our methodology for collecting unbiased peer uptimes.

Gnutella Cruiser: Cruiser employs several basic techniques and leverages features of modern Gnutella to significantly increase crawling speed as follows: First, Cruiser uses a special handshaking feature that is implemented by modern Gnutella clients. This feature enables the crawler to quickly query a peer for a list of its current neighbors. Previous crawlers relied on other features of the Gnutella protocol, namely Ping-Pong messages, to retrieve this information. These techniques were less efficient and potentially less reliable. Second, Cruiser leverages the two-tier structure of the modern Gnutella network by only crawling the top-level peers[1]. Since each leaf must be connected to an ultrapeer, this approach enables us to capture all the nodes and links of the overlay by contacting a relatively small fraction of all peers. Our recent measurements revealed that the degree of peer connectivity in the Gnutella network is high []. This significantly increases the rate of discovery for new peers and thus speeds up the crawling process because each contacted ultrapeer provides information about a large number of (potentially new) peers. Finally, Cruiser employs a master-slave architecture in order to achieve a high degree of concurrency and to effectively utilize available resources on multiple desktop PCs. A master process coordinates among multiple slave processes that act as virtually independent crawlers and crawl the network in parallel. To further improve the degree of concurrency, each slave process uses asynchronous communications to maintain hundreds of open connections in parallel.

These techniques collectively result in a significant increase in crawling speed. Cruiser can capture a snapshot of the Gnutella network with 300-400K peers *in less than 4 minutes* using 8 off-the-shelf 1GHZ Linux boxes in our lab. This is several orders of magnitude faster than previously reported crawlers (*i.e.,* 2 hours for 30K peers in [11], and 2 minutes for 5K peer in [4]). It is worth clarifying that while our crawling strategy is aggressive and our crawler requires a considerable amount of local resources, its behavior is not intrusive since each top-level peer is contacted only once per crawl. Note that there is a tradeoff between completeness and accuracy of captured snapshots. We examined this tradeoff and found that 2 minute crawls are optimal for Cruiser to capture the complete population of top-level nodes. For the remaining ultrapeers, the benefits of contacting them to find more peers is outweighed by the additional error the delay introduces. Further descriptions on Cruiser, required post-processing on each captured snapshot to remove inconsistencies, and performance evaluations of Cruiser can be found in our earlier work [12].

Unreachable Peers: In any arbitrary crawl, a significant portion (30%-38%) of discovered top-level peers are not directly reachable by a crawler. Previous studies assumed that unreachable peers have already departed and excluded them from their snapshots. We used the following post-processing strategy to determine the status of departed peers. Any peer that is unreachable during one crawl and does not exist in the next crawl, is considered departed during the first crawl. Our analysis show that only 2% of unreachable peers have departed during a crawl. Note that our goal is to determine (either directly or indirectly) the presence of any participating peer in each snapshot. Since the handshaking mechanism provides fresh and reliable information about neighbors of a contacted peer, we assume that all those peers are present in a snapshot even if they are not directly reachable. Therefore, error in our snapshots can only occur when a peer can not be discovered because it is only connected to unreachable peers. Given the high degree of connectivity among peers in the top-level overlay, the probability of such errors is very low. In fact, we have verified that our snapshots include the majority of peers in the top level overlay [12].

Collecting Unbiased Peer Uptime: To obtain a representative characterization of churn, it is essential to collect peer uptime in an unbiased fashion. Unfortunately, *this cannot be achieved without any priori knowledge about the distribution of peer uptime*. Since such information is not available, we take the following steps to minimize any error in our characterization. First, we capture complete snapshots of the Gnutella network to avoid any drawbacks of sampling. Second, to collect unbiased measurements of peer uptime, we have adopted the following methodology from Rosseli et al. in [13]: We divide our measurement period into two halves. Then, we only select the uptime for peers that satisfy the following three conditions: *(i)* join the system during the first half of our measurement, *(ii)* leave the system at any time during our measurement, and *(iii)* have an uptime shorter than half of our measurement period. This technique guarantees that the measurements are unbiased, but it provides us no information on peers with uptimes longer than half our measurement window. However, since our measurement window is sufficiently long, this is not a serious limitation.

III. Characterization of Churn

In this section, we present our preliminary results on the characterization of churn based on the above measurement methodology to collect peer uptime. Figure 1 presents variations in the population of top-level peers during one week (8/31/04 to 9/8/04) based on 5003

[1] Most modern Gnutella client implement a two-tiered overlay structure by dividing peers into two groups: ultrapeers (or superpeers) and *leaf* peers.

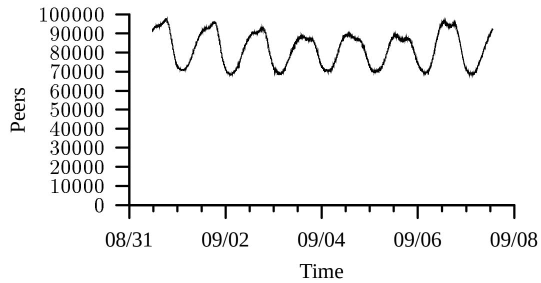

Fig. 1. Ultrapeer population over one week

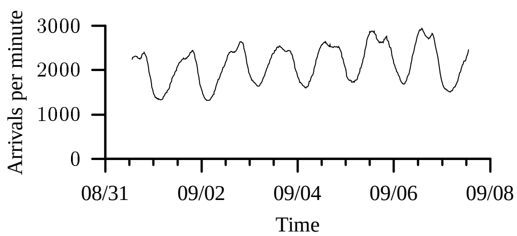

Fig. 2. Arrival/Departure rate over one week

back-to-back crawls. This figure clearly shows time-of-day effects with two spikes per day, corresponding to the east and west coast time zones of the US. To study the dynamics of peer participation, we examined peer arrival and departure rate. Figure 2 depicts the peer arrival rate for the same trace. Surprisingly, there is a strong correlation between peer arrival and departure rate (which is not shown here), *i.e.,* variations of departure rate follows the exact same pattern. This suggests that most peers have short uptimes.

Figure 3 depicts the distribution of peer uptimes among the top-level overlay from the same trace in log-log scale. This figure is based on 2,004,975 uptime measurements after applying our methodology to measure peer uptime, gathered from 2127 back-to-back 2-minute crawls. This figure clearly shows that uptime of top-level peers follows a power-law distribution. Using the least squares method on the log of the data, we fit this data to $p \propto e^{-1.75609*log(x)}$ which can also be expressed as $p \propto x^{-1.75609}$.

Since Figure 3 only presents the uptime of top-level peers, it does not capture two components: *(i)* the time were ultrapeers were not part of the top-level overlay, *(ii)* uptime for leaf peers that never become part of the top-level overlay. To address these shortcomings, Figure 4 presents the distribution of peer uptime based on 2,859,896 uptime measurements for all captured peers (including leaves), gathered from 1073 back-to-back

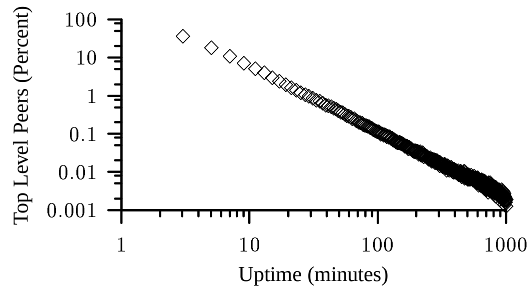

Fig. 3. Gnutella top-level uptime distribution peers up less than 24 hours, using bin width of 2 minutes, based on back-to-back crawls covering all of 9/1/2004 through 9/3/2004

crawls. Note that capturing the entire population takes longer, and reduces the granularity of measured uptime and thus we use a larger bin size. Using the least-square fitting approach, we have fit this figure into two different distribution as follows: *(i)* a two-piece power-law for x $<$ 118, $p \propto e^{-1.18356*\log x}$, for x $\geq$ 118, $p \propto e^{-1.89684*\log x}$, and *(ii)* a log-quadratic distribution $e^{-0.148933*(\log x)^2-0.125981*\log x}$. Further comparison between Figure 3 and 4 along with examination of various contributing components into these distributions remain as future work.

A. Implications of Power-law Distribution

The presented uptime distributions have several important implications as follows: First, this is a clear evidence that peer uptime does not follow the commonly used Poisson distribution. Second, in contrast to a Poisson distribution, the distribution of peer uptimes is not memoryless. More specifically, at any point of time, peer uptime is a good predictor for remaining uptime for that peer, *i.e.,* the longer a peer has been up, the

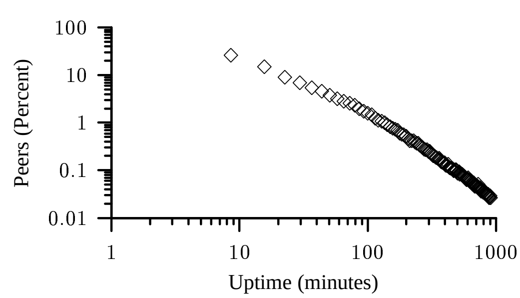

Fig. 4. Uptime distribution for all peers up less than 15 hours using a bin size of 7 minutes, based on data collected between 10/11/2004 1:45pm through 10/12/2004 8:13pm

longer it is likely to remain up. Third, most peers have short uptimes. However, most of the participating peers in any particular snapshot are long-lived peers. In a nutshell, a large portion of up peers are highly stable, yet the remaining peers turnover very quickly.

IV. Related Work

Several previous studies characterize P2P systems through measurement. Saroiu et al. [4] conducted a measurement-based characterization of several aspects of the Gnutella and Napster networks based on active measurement in May of 2001. They show the CDF of peer uptime indicating 50% of peers have uptime shorter than 60 minutes. They gathered their data by making short (2 minute), partial (25–50%) crawls to collect a list of hosts, then probing those hosts ate regular intervals. Chu et al. [5] conducted another measurement-based study on Gnutella and Napster in 2002. They showed that peer uptime follows log-quadratic distribution and 31% of peer have uptime shorter than 10 minutes. To collect their information, they have selected a group of 20,000 peers and contacted each peer once every ten minutes. As we discussed earlier, both these techniques result in biased characterizations.

Sen and Wang [7] used passive measurement to collect FastTrack traffic observed at backbone routers of a major ISP. To estimate peer uptime, they assumed that traffic from the same IP address with less than a 30 minutes gap are part of the same session, and concluded that 60% of the peers have uptime less than 10 minutes. Gummadi et al. [6] presented a study of Kazza based on passive measurement at a gateway of a university campus to the Internet. They reported that the median of observed peer uptime is only 2.4 minutes but did not provide any further information on the distribution of uptime. Due to the passive nature of measurements in these two studies, both the granularity and scope of their measurement are unknown. More specifically, a passive measurement can only detect a peer if it exchanges data with other peers, and that data traverses through the measurement point. Therefore, it is unclear whether the captured group of peers are representative. Bhagwan et al. [3] performed an active measurement study on a small group of Overnet peers (around 1500). Their goal was to examine "availability" of a specific peers. Using Overnet network enabled them to identify individual peers any time they join the system even if they use different IP addresses due to DHCP or a firewall. However, to examine churn, we are primarily interested in the length of each single participation period (from arrival to subsequent departure) by each peer. Therefore, the total uptime of each peer over the measurement period can not be directly used to characterize churn.

Finally, Nowell et al. [9] introduced the notion of a "half-life" to model churn in P2P networks and examine its impact on DHT performance. Their primary assumption is that peers arrive according to a Poisson distribution. Other studies use such a model to evaluate their P2P applications [2]. Our characterizations reveals that this assumption is not correct. *We are not aware of any previous measurement study on P2P networks that captures complete snapshots of a large P2P network over sufficiently short time scales to properly characterize churn.*

References

[1] Y. Chawathe, S. Ratnasamy, and L. Breslau, S. Shenker "Making Gnutella-like P2P Systems Scalable," in *SIGCOMM*, 2003.

[2] S. Rhea, D. Geels, and J. Kubiatowicz, "Handling Churn in a DHT," in *USENIX*, 2004, pp. 127–140.

[3] R. Bhagwan, S. Savage, and G. Voelker, "Understanding availability," in *IPTPS*, 2003.

[4] S. Saroiu, P. Gummadi, and S. Gribble, "Measuring and Analyzing the Characteristics of Napster and Gnutella Hosts," *Multimedia Systems Journal*, vol. 8, no. 5, Nov. 2002.

[5] J. Chu, K. Labonte, and B. Levine, "Availability and Locality Measurements of Peer-to-Peer File Systems," in *ITCom: Scalability and Traffic Control in IP Networks*, July 2002.

[6] K. Gummadi, R. Dunn, S. Saroiu, S. Gribble, H. Levy, and J. Zahorjan, "Measurement, Modeling, and Analysis of a Peer-to-Peer File-Sharing Workload," in *SOSP*, 2003.

[7] S. Sen and J. Wang, "Analyzing Peer-To-Peer Traffic Across Large Networks," *IEEE/ACM TON*, vol. 12, no. 2, Apr. 2004.

[8] J. Li, J. Stribling, T. Gil, R. Morris, and F. Kaashoek, "Comparing the performance of distributed hash tables under churn," in *IPTPS*, 2004.

[9] D. Liben-Nowell, H. Balakrishnan, and D. Karger, "Analysis of the Evolution of Peer-to-Peer Systems," in *Principles of Distributed Computing*, Monterey, CA, July 2002.

[10] R. Mahajan, M. Castro, and A. Rowstron, "Controlling the Cost of Reliability in Peer-to-Peer Overlays," in *IPTPS*, 2003.

[11] 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 Computing Journal*, vol. 6, no. 1, 2002.

[12] D. Stutzbach and R. Rejaie, "Evaluating the Accuracy of Captured Snapshots by Peer-to-Peer Crawlers," Tech. Rep. CIS-TR-04-05, University of Oregon, Nov. 2004.

[13] D. Rosseli, J. Lorch, and T. Anderson, "A Comparison of File System Workloads," in *USENIX*, June 2000.