Churn Impact on Replicated Data Duration in Structured P2P Networks

Xu Guangping[1,2], Ma Wenhui[1] Wang Gang[1], Liu Jing[1]

1. Information Technology Science College, Nankai Universitity, Tianjin, China
2. Computer Science and Technology School, Tianjin University of Technology, Tianjin, China
Email: xugp2008@yahoo.com.cn

Abstract

This paper analyzes churn impact on replicated data duration with different node lifetime distributions. In structured overlay networks, churn includes node-join churn and node-failure churn, caused by the arrival and departure of nodes separately. The paper introduces a duration model of replicated data under node-failure churn for node failure directly leads to data loss. Furthermore, it investigates the impact of node-join churn on the duration of replicated data for different node-lifetime distributions. The paper presents that node-churn will negatively impact on replicated data duration for heavy-tailed distribution and Weibull distribution except exponential distribution. Then we evaluate the impact on replicated data duration with two real-world trace datasets. The experimental results show the negative impact of node-join churn for different node-join churn degrees. Finally, the paper discusses an enhancement by setting a trial period for every fresh node. By experiment, it is an effective way to reduce the negative impact of node-join churn due to the memory property of node lifetime distributions.

1. Introduction

Structured overlay network is a high logical level network architecture built on the existing networks, depending on the distributed hash table (DHT) infrastructure to store data objects. As shown in the recent research, the approach can supply resilience to node failures, scalability, and location independence using only local routing information. However, various DHTs don't offer preferable guarantees about data availability for the networks are commonly faced with high rates of churn, i.e., nodes join and fail frequently in systems. In the large scale systems, the preserved increasing data collections are the base supports for upper-level services and applications. To efficiently share and preserve data, data availability is the fundamental issue expected to solve in these systems.

To achieve high data duration, there are at least the following required aspects. First, the churn characteristics should be understood for the large scale systems. Second, data redundancy should be employed properly in the systems. Motivated by the aspects, this paper focuses on the analysis of churn impact on replicated data duration in structured P2P networks. The paper provides a duration model of replicated data under node-failure churn according to reliability theory. The model mainly involves the number of replicas and node lifetime distribution. However, it ignores the impact of node-join churn in the model. Even though node-join churn does not lead to data loss, it causes data migration in structured P2P networks. The data migration may change replicated data duration for different replica placements. Therefore, node-join churn may impact on replicated data duration indirectly. The paper analyzes the impact of node-join churn on data duration for different node lifetime distributions. And then the evaluation verifies the negative impact of node-join churn on the replicated data duration based on two empirical trace datasets. Moreover, the paper discusses the enhancement by setting a trial time for every fresh node to reduce the negative impact.

The rest of the paper is organized as follows. Section 2 briefly introduces the related work. Then in the next two sections, our paper presents replicated data duration model and reveals that the implicit reason of data unavailability is node-join churn for different node lifetime distributions. In section 5, we evaluate the above analysis including lifetime characteristics and the negative impact of node-join churn. In section 6, the paper discusses an enhancement.

2. Related work

Some widely-deployed internet systems, such as PlanetLab [6], Skype [21], provide a platform to study dynamics of peers in the large-scale real environment. For a node, a cycle time of a node from its join till its leave is called a session time. In this paper it is called node lifetime as in reliability theory. Replicas are stored in different nodes. The replicated data duration is the convergence lifetimes of the nodes replicas stored in [1-3,8,21]. Node lifetime distribution is indispensable to analyze replicated data duration. By measurements in these real-world systems, some observations about node lifetime distributions were proposed. But the observations

are different in real different systems. For example, heavy-tailed distribution was adopted in [11,12] and Weibull or lognormal distribution was employed in [13] while some still took it as exponential distribution in [9,10].

Leonard et al [12] studied about the isolated node lifetime, which gives us some inspiration on data duration model. In paper [5], it presents the guides to minimize churn by different node selection strategies. However, the related work focuses on network resilience to improve the node duration rather not on the resided data. These efforts facilitate our work on data duration under churn. Based on referencing the model and trace observations, we can analyze and evaluate replicated data duration under churn.

In DHTs, replica placement strategies were discussed in [8,5], classified as root set scheme and random scheme. For any placement scheme, churn may lead to replica migration among related nodes and furthermore the migration may change the data duration. Some work [14,15] studied the availability relationship among the mean availability of peers, the number of replicas and the required data availability. But they assumed each replica could not migrate to other node. Therefore, they ignored the impact of churn, especially node-join churn. Moreover, replication maintenance is an indispensable part to achieve high data availability [16,18,19].

3. Duration model

3.1. Replica placement

We assume that an item initially is duplicated into m replicas. The number of replicas of item at any time is called replication factor (*f*). An item is called alive if *f>0*; otherwise, an item will be lost if *f=0*. The period of replication factor of an item from *m* to *0* is called the duration of an item.

For managing replicas in structured P2P networks, two common replica placement strategies, *root set* and *random*, are employed for replica placement, as stated in [5]. In root set strategy, the node whose key most closely follows k serves as the owner of an item which key is *k*, and replicates the item into (*m-1*) neighbors. In random strategy, the responsible node of each replica is determined by a set of known re-hash functions in the global space. According to the strategies, the selected nodes fully depend on consistent hash functions and these nodes have existed for random amount of time before the item is put into a system. That is, the selection does not concern about the residual lifetimes of the resided nodes in the system. Therefore, all replicas randomly reside on different nodes from the nodes already present in the networks when an item is put into a system. Thus it guarantees that node random selection for replicas is independent of node lifetimes and their current ages.

3.2. Model

Consider an item of which each replica for a variable length of time having a distribution function $F_i(t)$ and then fails [7]. After a length of time t has elapsed, the probability that each replica remains alive is given by

$$\bar{F}_i(t) \equiv P\{\text{replica lifetime} > \text{t}\} = 1 - F_i(t). \quad (1)$$

Therefore, the probability that an item will be alive for a length of time t or greater is given by

$$\bar{F}(t) = r(\bar{F}_1(t), ..., \bar{F}_m(t)) = 1 - \prod_{i=1}^{m} F_i(t). \quad (2)$$

At time t when an item is put into the system, each replica is fully random resided on a node and t is uniformly random within each node life. Therefore, the replica lifetime is equivalent to the residual lifetime of the resided node. According to [7], the residual lifetime of each node, that is the lifetime of each replica is given by

$$R(t) = \frac{1}{E[L]} \int_0^t (1 - H(z)) dz. \quad (3)$$

where $H(t)$ is the distribution function of each node lifetime L and $E[L]$ is the expected value. The probability of each replica alive can be obtained by the formula at any time t. Assuming that m replicas of an item are independent, and each of which lifetime distribution function is $F_i(t)=R(t)$, then the expected lifetime of an item is derived as follows.

$$\begin{aligned} E(T) &= \int_0^\infty P\{L > t\} dt = \int_0^\infty \bar{F}(t) dt \\ &= \int_0^\infty \left(1 - (R(t))^m\right) dt \\ &= \int_0^\infty (1 - (\tfrac{1}{E[L]} \int_0^t (1 - H(z)) dz)^m) dt \end{aligned} \quad (4)$$

As result, we obtain the formula (4). It gives the expected time interval before all the resided nodes fail. The result shows the expected duration of an item, E[T], is the convergence the residual lifetimes of all selected nodes, and it is dependent on the distribution of node lifetime, H(t), and replication factor, m. The model represents the expected convergence duration of originally selected nodes for replicas until all the nodes fail. However, each replica of an item may not reside on the originally selected node till the node fails in fact. That is, replicas may migrate to other nodes. What causes the replica migration? How does the replica migration influence on the model? In next section, we analyze these problems.

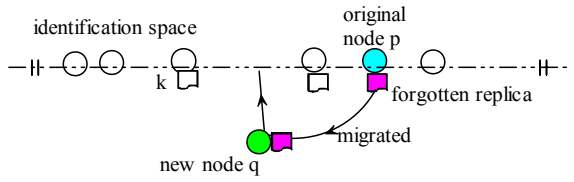
4. Analysis

4.1. Impact analysis

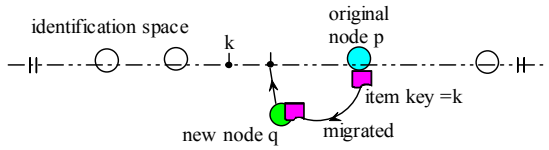
It is inevitable that some replicas may migrate to other nodes during their lifetimes. As we know, when a new node joins in structured P2P networks, the related data

may be migrated to the new node from its original one in the both placement strategies, *root set* and *random* replica placement strategies. Figure 1 shows that the replica migration happens from the original node *p* to the new node *q* for an item *k* for the both placement strategies. When the new node *q* joins which is closer than node *p* for item *k*, it replaces the original node *p* and takes the responsibility for a replica of item *k*.

In the process, the related data is emigrated to it from the original node in a proactive way when a new node joins the systems. The data migration may impact data duration even though node-join churn does not cause data lost.



(a) root set placement strategy



(b) random placement strategy

Figure 1. Data migration due to node-join churn

The proposed model only considers node-failure churn from originally selected nodes and ignores node-join churn from new joined nodes. In fact, node-join churn is inevitable and brings impact on the data duration.

In figure 2, an item is put into systems at time *t* and one replica of an item is replicated to the responsible node *p*. At time *s*, the replica migrates to a new node *q* from the original node *p*. Let R_t denote the residual lifetime at time *t* of a node representing an interval of time till the node fails and let *L* denote the overall lifetime of new node. Intuitively, it seems that the lifetime *L* of a new node is larger that the residual lifetime R_s of the replaced node after time *s*. But the intuition is wrong since the result is dependent on the node lifetime distribution.

As known from [7], the comparison between the expected lifetime (*L*) and the expected residual lifetime (*R*) uniquely lies on the failure rate of node,

$$z(t)=h(t)/R(t) \qquad (5)$$

where *f(t)* is the density of *L* . The failure rate function is an indicator of the proneness to failure of the node after time *t* has elapsed. If *z(t)* is a decreasing function of *t*, then the node lifetime distribution is a decreasing failure rate distribution. Therefore, the impact of node-join churn depends on the failure rate of node lifetime which is a decreasing or increasing function.

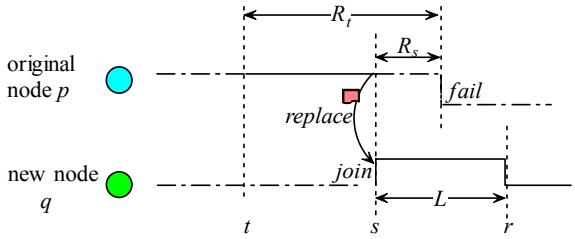


Figure 2. The relationship between E[L] and E[R].

4.1.1. Exponential distribution. Some prior work [9,10] employed exponential distribution to study churn in P2P networks for simplicity. For the exponential distribution of node lifetime, $H(t)=1-e^{-\lambda t}, \lambda>0, t\geq 0$. Derived from (3), the residual lifetime distribution is the same as the lifetime distribution, *H(t)= R(t)*. Therefore, *E[L] equals to E[R]*. It can be also explained by the memory-less property of exponential distribution, i.e., a node does not age with time whether it is replaced or not. Therefore, it can be summed up that node join churn can't influence lifetimes of data items for exponential distribution.

4.1.2 Pareto distribution. Currently, some studies [11,12] revealed that the lifetimes of nodes in P2P systems exhibit the heavy-tailed characteristic. As a typical heavy-tailed distribution, Pareto distribution can allow arbitrarily small lifetimes to represent node lifetimes in systems, with the distribution function given by $H(t)=1-(1+\frac{t}{\alpha})^{-k}$, *t>0*. For the distribution, the mean lifetime of a new node is *E[L]= a/(k-1)*. The residual lifetime distribution of the node replaced by the new joined node obtained from (3) is $R(t)=1-(1+\frac{t}{\alpha})^{1-k}$ and the expectation is *E[R]= a/(k-2)*. Therefore, the comparison result is *E[L]<E[R]*. From the point of view of the failure rate, *z(t)=k/(a+t)*, which is monotonically decreasing as a function of *t*, we also obtain the same result. Therefore, Pareto distribution exhibits a strong memory property. If the time interval *(s-t)* approximates to 0, then *L* tends to equal *R*. As the interval (*s-t*) increases, the inequality *L<R* holds much more. That is, the larger the current age of a peer, the longer it is expected to remain online. If a node is fresh, then its life may be short to some degree. Therefore, it can be summarized that node-join churn negatively influence the duration of replicated data for heavy-tailed distribution.

4.1.3. Weibull distribution. Other observations suggest that the lifetimes are neither exponential distribution nor heavy-tailed distribution, but Weibull distribution [13]. For Weibull distribution, $H(t)=1-\exp[-(\lambda t)^{\alpha}], t\geq 0$, the mean and the squared coefficient of variation are $E(L)=\frac{1}{\lambda}\Gamma(1+\frac{1}{\alpha})$ and $z(t)=a\lambda(\lambda t)^{a-1}$ respectively.

Therefore the shape parameter, $0<a<1$, ensures that the failure rate is decreasing. As shown in [13], the Weibull distribution provided a tighter fit with shape parameters (a) in three different systems are 0.34, 0.38 and 0.59 separately. So for these distribution, $E[L]<E[R]$. Therefore we can safely summarize that node-join churn negatively influence lifetime of data items for the Weibull distribution with shape parameter $a<1$.

All in all, we summarize that node-join churn negatively impact on the duration of replicated data. The impact is coming from the memory property of node lifetime distribution, Pareto distribution and Weibull distribution, in most existing measurements. The replicated duration in the model can be taken as the upper bound of the duration of replicated data approximately.

5. Evaluation

To evaluate our analysis of replicated data duration, we developed a trace-driven simulator. The used datasets come from two real-world traces, PlanetLab [2, 5, 6] and Skype [4] trace datasets. The trace datasets of these systems are collected over a long term and publicly available [22].

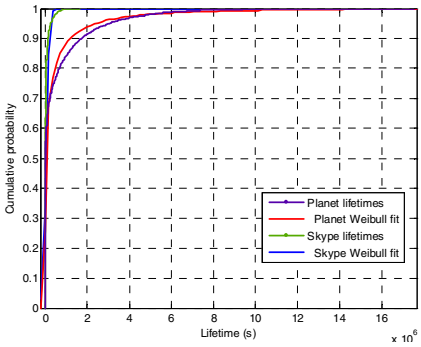


Figure 3. Node lifetime distributions in the trace datasets

The trace datasets contain a list of time intervals for which each node was online contiguously. PlanetLab is a large-scale, distributed testbed with nodes located around the world. PlanetLab trace dataset recorded the living status of 669 nodes in about 18 months ($4.557*10^7$ s) which was collected by the CoMon project [21]. The other trace dataset was got from Skype system with 2081 nodes in about one month ($2.4782*10^6$s). Guha et. al. [4] presented a study of peer behavior in the Skype system. We utilized these system traces as well-suited tools to study churn characteristics of large-scale distributed P2P platform. In figure 3, it shows that node lifetimes in the traces fit tightly to Weibull distribution at a 95% confidence interval. For the two fits, the Weibull distribution parameters (a, λ) are (0.378, 133268) for PlanetLab trace and (0.644, 39108.5) for Skype trace separately. Therefore, as analyzed in section 4.3, the failure rates are decreasing due to the shape parameters $a<1$. In the next subsection, we can verify that node-join churn negatively influence lifetime of data items for the distribution by our experiments.

In the experiments, we evaluated the duration of replicated data and the impact of node-join churn on the duration. In our evaluation, the following node sets are randomly generated. Replica node set (*RS*) is defined as the chosen nodes to store replicas of a data item. Joined node set (*JS*) is defined as the chosen nodes to join the system in a period.

First, we made the comparison between the expected lifetime of the fresh node and the expected residual lifetime of replaced node after the fresh node joins. In an experiment, the nodes of *RS* are randomly selected from all living nodes at a sample time clock. The size of *RS* is equal to replication factor ($m=10$). During the duration of *RS*, the nodes of JS are determined. One node is randomly chosen among *JS* to replace a living node in RS. Thus, the lifetime of a fresh node and the residual lifetime of a replaced node are obtained. The sample interval is fixed and sampled times is 11 in one experiment during the trace. The above procedure is repeated 20 times and the average lifetime ($E[L]$) and the residual lifetime ($E[R]$) are got by the experiments. The results are shown in figure 4. In the figure, the X-axis is discrete samples and the Y-axis is the expected lifetime of nodes. It clearly shows that $E[R]$ is greater that $E[L]$. Therefore, the observation verifies the analysis in section 4.3.

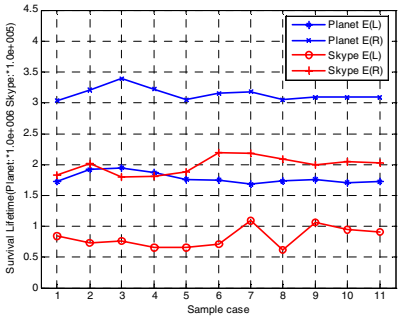


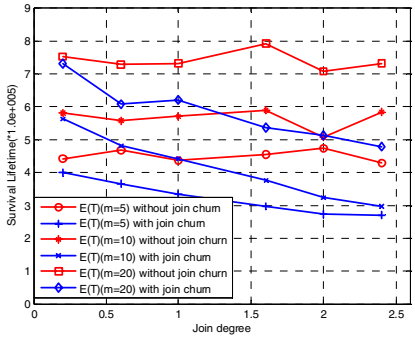
Figure 4. The comparison between the expected lifetime of a fresh node and residual lifetime of replaced node.

Next, we evaluated the impacts of different degrees of node-join churn on the duration. To evaluate the impacts of different extents of node-join churn, we define a metric *join churn degree*,

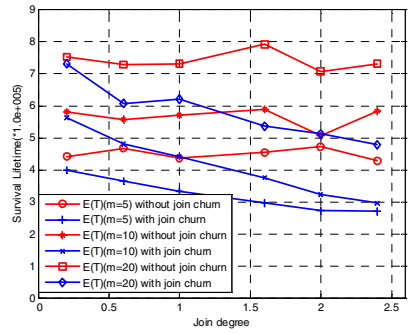
$$j=s*E[T]/m \quad (6)$$

where s presents node join rate relative to each living peer and $E[T]$ presents the expected duration of m replicas without node-join churn. Thus the product of s and $E[R]$ is the number of nodes joining the system during the period $E[R]$. The metric j presents the ratio of the number of joined nodes to the size of *RS* during the replicated data

duration. Therefore, in the experiments the size of *JS* is set the value *j***m*.



(a)



(b)

Figure 5. The expected durations for different parameters with/without node-join churn

According to the given parameters *m* and *j*, we evaluated the impact of node-join churn on replicated data duration. *RS* firstly was built and then during the duration of *m* replicas without node-join churn, a number of joined nodes (*j***m*) were selected to replace the resided nodes, that is, *JS* is randomly generated. Every node in *JS* joined the system in the trace data time order and randomly replaces an existing node in *RS*. If a selected node to be replaced has failed when a node joins the system, then the join was ignored. Therefore the replicated data duration was obtained with node-join churn in the simulation.

In the experiments, the expected durations were obtained for 2000 items for the different replication factors: *m*=5, 10, and 20. We ran the experiments without and with node-join churn. With node-join churn, different churn degrees were set, *j*=0.2, 0.6, 1, 1.6, 2 and 2.4. Figure 5 shows the experimental results. In each plot, the X-axis indicates the node-join churn degrees and the Y-axis indicates the replicated durations. From the plots, for each *m* without join churn, the expected duration fluctuates about a constant duration. However, for each *m* with join churn, the duration curve is below the curve without join churn and the curve with greater node-join churn degrees is a downward trend, i.e., the average duration decreases as the join-churn degree increases.

In summary, our evaluation results indicate the following observations. Obviously, with a larger replication factor (*m*), the replicated data duration is larger. Even though increasing replication factor can improve the duration, it would bring some overheads, such as storage space and replication maintenance bandwidth. It's worth noting that the expected duration of replicated data is negatively impacted by node-join churn. When node-join churn occurs frequently (the higher value *j*), the negative impact is more evident. Next section discusses replication maintenance to reduce the negative impact of churn, especially node-join churn impact.

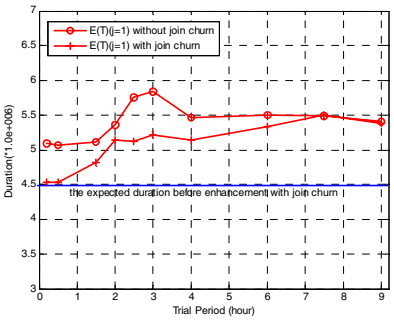
6. Discussion

According to the above model and analysis, heavy-tailed distribution and Weibull distribution have memory property that guarantees that node future lifetime could be predicted based on the current age of node [14]. Therefore, it is a preventive way to set a trial period for every fresh node to reduce the negative impact of node-join churn.

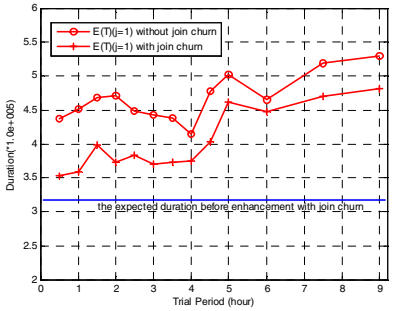
We evaluated the effects of different trail periods by simulation using the traces as section 5. In the experiments, every fresh node is observed and determined whether it should replace an existing node for some related replicas. During the trial period, the fresh node is inspected and reserve of its responsible data. After the trial period, it becomes the formal node in networks to serve its responsible data. Different replication strategies in the trial period could be chosen between the fresh node and the original node. Whether does the fresh node serve data service during the period? When does data migration happen? In our simulation, the fresh node joins the system, replicates data from the original node and provides routing and data service in a normal way of structured overlay network requirement during the trial period. During the period, the original node could provide the data voluntarily. After the period, the original node could not hold the replicated if the fresh node is still alive.

We evaluated the average durations by setting different trial periods under the parameters: j = 1and m=5 for 2000 items. To eliminate the churn impact from an amount of short-lived peers (as shown in table II), different trial periods were set in the simulation. Figure 6 shows that the enhanced curves with node-join churn are lower than the ones without node-join churn. Clearly, the enhanced duration is larger than the durations with node-join churn before enhancement. Figure 6 also shows that the average durations increase with increasing trial periods. Therefore, the enhancement reduces the negative impact of node-join churn, and extends the durations of replicated data. However, the trial period setting is need to tradeoff between storage space and the expected data duration. If the trial period is too large, then the storage space would be need more and much maintenance workload would be imposed on some nodes. If the period

is too small, the extending effect on the duration is not obvious. So a moderate length of the trial period can be decision, e.g., 2 hours for PlanetLab and 5 hours for Skype are the proper values.



(a)PlanetLab



(b)Skype

Figure 6. The expected survival duration of data item by setting different trial period to joined nodes

ACKNOWLEDGEMENTS The work in part is sponsored by National Science Foundation of China (No.90612001), Education Ministry Doctoral Research Foundation of China (No.20070055054), Science and Technology Development Plan of Tianjin (No. 043185111-14).

References

[1] Frank Dabek, M. Frans Kaashoek, David Karger, Robert Morris, and Ion Stoica. "Wide-area cooperative storage with CFS," in Proc. of the 18th ACM Symposium on Operating Systems Principles (SOSP'01), Chateau Lake Louise, Banff, Canada, October 2001, pp.202-215.

[2] Byung-Gon Chun, Frank Dabek, Andreas Haeberlen and el. "Efficient replica maintenance for distributed storage systems," in Proc. of the 3rd Symposium on Networked Systems Design and Implementation. San Jose, CA, May 2006, pp.45-58.

[3] R. Bhagwan, K. Tati, Y. Cheng, S. Savage, G. Voelker. "Total recall: System support for automated availability management," in Proc. of ACM/USENIX NSDI'04, San Francisco, California, March 2004, pp. 337-350.

[4] Saikat Guha, Neil Daswani and Ravi Jain. "An Experimental Study of the Skype Peer-to-Peer VoIP System," in Proc. of the 5th International Workshop on Peer-to-Peer Systems, Santa Barbara, CA, February 2006, pp. 1-6.

[5] Brighten Godfrey, Scott Shenker and Ion Stoica.. "Minimizing Churn in Distributed Systems," in Proc. of ACM SIGCOMM'06, Pisa, Italy, September 2006, pp147-158.

[6] Jeremy Stribling. Planetlab all pairs ping. [Online]. Available: http://infospect.planetlab.org/pings.

[7] Terje Aven, Uwe Jensen. Stochastic models in reliability, New York: Springer, 1999.

[8] Kiran Tati and Geoffrey M. Voelker. "On object maintenance in peer-to-peer systems," in Proc. of the 5th International Workshop on Peer-to-Peer Systems, Santa Barbara, CA, February 2006.

[9] D. Liben-Nowell, H. Balakrishnan, and D. Karger. "Analysis of the Evolution of Peer-to-Peer Systems," in Proc. of the 21st ACM Symposium on Principles of Distributed Computing, Monterey, CA, July 2002.pp.233–242.

[10] S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz. "Handling Churn in a DHT," in Proc. of the USENIX Annual Technical Conference, Boston, MA, USA, June 2004, pp.127–140.

[11] Saroiu S, Gummadi PK, Gribble SD. "A measurement study of peer-to-peer file sharing systems," in Proc. of the SPIE/ACM Conference on Multimedia Computing and Networking 2002, San Jose, CA, January 2002, pp.156-170.

[12] D. Leonard, V. Rai, and D. Loguinov. "On Lifetime-Based Node Failure and Stochastic Resilience of Decentralized Peer-to-Peer Networks," in Proc. of ACM SIGMETRICS, Banff, Canada, Jun. 2005, pp. 26-37.

[13] D. Stutzbach and R. Rejaie. "Understanding Churn in Peer-to-Peer Networks," in Proc. of the 6th ACM SIGCOMM on Internet measurement, Rio de Janeiro, Brazil, October 2006, pp.189-202.

[14] R. Bhagwan, S. Savage, and G. Voelker. "Understanding Availability," in Proc. of the 2nd Int'l Workshop Peer-to-Peer Systems (IPTPS 05), Berkeley, CA, February 2003, pp.256-267.

[15] Rodrigo Rodrigues and Barbara Liskov. "High availability in dhts: Erasure coding vs. replication," in Proc. of the 4th International Workshop Peer-to-Peer Systems (IPTPS 05), Springer, 2005, pp. 226-239.

[16] Sit, E., Haeberlen, A., Dabek, F., Chun, B.-G., Weatherspoon, H., Morris, R., Kaashoek, M. F., and Kubiatowicz, J. "Proactive replication for data durability," in Proc. of the 5th Int'l Workshop on Peer-to-Peer Systems (IPTPS 06), 2006.

[17] W.K. Lin, D.M. Chiu, and Y.B. Lee, "Erasure code replication revisited," in Proc. of the 4th International Conference on Peer-to-Peer Computing, , Aug. 2004, pp.90–97.

[18] Dabek, F., Li, J., Sit, E., Rrbertson, J., Kaashoek, M. F., and Morris, R. "Designing a DHT for low latency and high throughput," in Proc. of the 1st Symposium on Networked Systems Design and Implementation (NSDI'04), San Francisco, California, March, 2004, pp.85-98.

[19] Haeberlen, A., Mislove, A., and Druschel, P. "Glacier: Highly durable, decentralized storage despite massive correlated failures," in Proc. of the 2nd Symposium on Networked Systems Design and Implementation (NSDI'05), Boston, MA, May 2005.

[20] Guangping Xu, Gang Wang, Jing Liu. A hybrid redundancy approach for data availability in structured P2P network systems, in Proc. of the 13th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC'07) , Melbourne, Victoria, AUSTRALIA, December, 2007.

[21] Park K. S., and Pai, V. CoMon: a mostly-scalable monitoring system for PlanetLab. ACM SIGOPS Operating Systems Review 40, 1 Jan. 2006, 65–74.

[22] Trace dataset package. [Online]. Available: http://www.cs.berkeley.edu/~pbg/availability/.