

# Understanding the Dynamic of Peer-to-Peer Systems\*

Jing Tian, Yafei Dai  
CNDS Lab, Peking University  
{tianjing, dyf}@net.pku.edu.cn

## Abstract

*Though a few previous research efforts have investigated the peer availability of P2P systems, the understanding of peer dynamic is far from adequate. Based on the running log of a file-sharing P2P system, we produced a more thorough measurement of the dynamic natures of P2P system. We further show that due to the methodology limitation, crawler based measurement can not precisely capture system dynamic natures as a whole. In this paper, we also emphasize some simple yet important dynamic metrics, which are omitted or neglected by previous studies because of state of the art in durability analysis at that time. By a fine-grained analysis on the preliminary findings, we reveal a series of useful implications for the design of P2P systems.*

## 1. Introduction

P2P systems have the potential to be failure resilient because each peer functions equally and does not rely on any central server. Nevertheless, every peer is meant to join and leave the system arbitrarily, which makes failure more common than in other systems. The designers of critical applications such as the P2P storage system must know the dynamic natures to develop mechanisms masking the transient failures as well as the permanent failures. Even the designers of non-critical applications should know about the dynamic natures to optimize the system performance, e.g. how often to republish the indices in a file-sharing system.

Though very important, the understanding of the dynamic nature are far from adequate compared with mature research fields [1], for two reasons: first, it is not a trivial work to control a whole P2P system and give a thorough measurement; secondly, in this emerging area, the space of potential applications is still poorly understood, so it is very difficult to judge whether a particular measurement is representative of an entire class of applications. In this paper we contribute to the community more first-hand measurement results to supplement existing works by analyzing the entire running log of Maze system [2].

To build a real deployed popular P2P application is difficult, tedious and time-consuming work, so previous measurement studies usually use a crawler to crawl a fraction of hosts as a snapshot, and periodically probe their availability. This methodology inherently sets a great barrier to our understanding of the dynamic from the whole system-wide view and from the long-term evolution view. Particularly, the long-term lifetime has been becoming more and more important in the research of maintenance bandwidth for P2P storage systems[3, 4]. In this study, we use the entire system running log to characterize the whole system-wide dynamic behaviors as well as the long-term evolution natures. By comparison, we find that crawler based measurements dramatically underestimate the system dynamic. Based on the long-term measurement, we find that the newly registered peers generally have much higher turnover rate than that of elder ones.

Another limitation of previous measurements is the lack of some important metrics due to the state of the art of durability analysis. For example, at the beginning of P2P storage research, combinatorial probability computation dominates the computation of object availability evaluation [5] so the measurements concentrate on host availability. However, the basic parameter of subsequent stochastic analysis is peer session time distribution which is poorly studied by comparison. Another rising discussion is to use extra replicas masking transient failures [5-7], the key for this being false positive probability of permanent failure detection is also neglected. In this work, we give a thorough measurement for all these metrics beyond availability.

This study was initially motivated by a durable P2P storage research [8]. For a storage system, we may improve the performance by using low dynamic peers as super peers. Consequently, we give a fine-grained analysis by clustering peers according to their dynamic, and some of the conclusions are far different from the ones from all peers set.

The body of this paper is organized as follows. After a survey of related works in section 2, we briefly introduce the Maze system and our methodology in section 3. In section 4, measurement results as well as the preliminary analysis are reported, including both short-term and long-term dynamic natures. Finally, section 5 summarizes this paper.

---

\* This work is supported by National Grand Fundamental Research 973 program of China under Grant No.2004CB318204, National Natural Science Foundation of China under Grant No.90412008

## 2. Related Works

Saroiu et al. [9] were one of the first to perform detailed measurements of Napster and Gnutella by using crawlers and probers, and reported the system dynamic natures. Sen et al. [10] measured the dynamic of P2P systems by analyzing flow-level data at a large ISP. In a closely related effort, Bhagwan et al. [11] have studied the availability of Overnet by probing the crawled hosts and have shown that the methodological limitation of IP based measurement in previous works had dramatically underestimated host availability. In [12], Qiao et al. reported a similar dynamic level in Gnutella and Overnet. Guha et al. [13] found Skype to have much higher host availability than other P2P systems. Unfortunately, all of these results are only from a crawled peer set, a fraction of all peers in system. We will now show how this methodology underestimates the dynamic.

A number of measurement works on less dynamic distributed systems are also available. [14] characterizes the dynamic of large distributed systems by analyzing three probing traces. [15, 16] studied the dynamic of desktops in enterprise environments. In this paper, we focus on the dynamic natures of wide-area P2P systems.

The measurement studies are driven by the requirements of durability analysis and evaluation researches. [3, 5, 17] evaluate data availability by combinatorial probability computation which requires the host availability as parameter. [7, 18-20] instead conduct their evaluations on stochastic process models, and thus session time and its distribution are their key parameters. Recently, the discussion on lazy repair [5-7, 21] by using extra replicas is coming up in P2P storage community. The number of extra replicas relies on the false positive probability of permanent failure detection determined by the detection threshold and transient offline time distribution, which are also neglected in previous measurements. [3] addresses that peer departure rate is important, which determines the feasibility of building the high available P2P storage systems. We highlight these omitted metrics in our study, and give our preliminary analysis results.

## 3. Maze Background and Methodology

Our measurements are based on the running log dataset of Maze [2], a P2P file sharing system developed, deployed and operated by our academic research team. Maze has a cluster-based central service for resource searching and peer activities logging. Maze now is one of the largest P2P systems over CERNET (China Education and Research Network), with an average of 20K simultaneously online users. This large population gives us reason to believe that Maze is an excellent platform for

measurement studies. Based on Maze log, a number of measurement results [2, 22, 23] have been reported.

In Maze, peers are identified by sequentially generated immutable IDs rather than IP addresses, thus eliminating the host aliasing problem caused by the widely used DHCP and NAT. Online peers periodically send heartbeats to claim their existences, while log server makes a snapshot of online users every 5 minutes. By concatenating the snapshots, we can have a whole system view of peer dynamic. We conduct our analysis on the system log from 3/1/2005 to 5/31/2005. During this period, more than 500K peers participated in system at least once. We have lost the snapshots from 4/1 to 4/3 and 5/1 because of operating accidents. Nevertheless, this does not affect the outcome of analysis.

## 4. Measurement Results

In this section, we present the results of our measurements and analysis as well as their implications for system design and evaluation. The results are divided into two parts, the short-term dynamic and long-term dynamic. Short-term dynamic natures mainly depict transient peer failures that may affect the availability of archived data or the durability of temporarily stored states such as keyword indices in file sharing systems. On the other hand, long-term dynamic natures reflect the evolution of the system and impact the reliability of archived data.

### 4.1. Short-term Dynamic

#### 4.1.1. Availability and Bias of Crawler Based Measurement

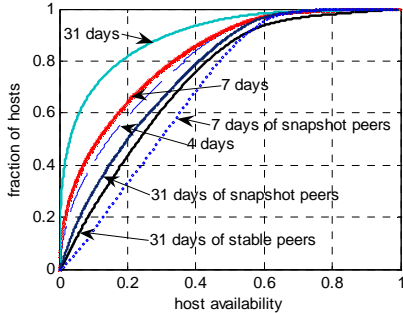
Host availability is the traditional dynamic metric of P2P systems. In this subsection, we first report the peer availability in Maze, then shed light on the methodological bias of crawler based measurement. Since we are focusing on short-term dynamic, we will only conduct our analysis on March's logs in the following short-term measurements.

Noting that availability distribution varies with the monitoring time period [11], we plot the cumulative distributions of availability over first 4 days, first 7 days and all 31 days in March in **Fig1**. Obviously, availability over 31 days is much worse than the ones over 7 days and 4 days, while the one over 4 days is the best.

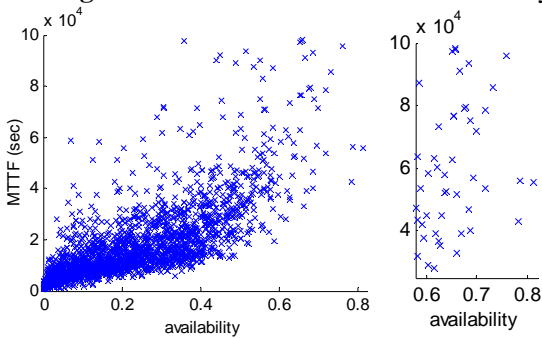
To compare our availability distribution with the crawler based ones, we take a snapshot of all 21,399 online peers at 10:00pm, 3/1/2005. **Fig1** also depicts availability distributions of these peers over first 7 days and all 31 days. Clearly, availabilities of snapshot peers are much better than those of all peers. We believe the reason is that crawler tends to capture peers with high availability because they are more likely to be online. As a result, crawler

based measurement may greatly underestimate system dynamic.

For snapshot peers over 7 days monitoring, over 60% peers have availability over 0.2, and about 30% peers have availability over 0.4. These show that dynamic of Maze is close to Overnet [11].



**Fig1.** Cumulative distribution of availability



**Fig2.** Availability vs. Mean Session Time

#### 4.1.2. Availability vs. Mean Session Time

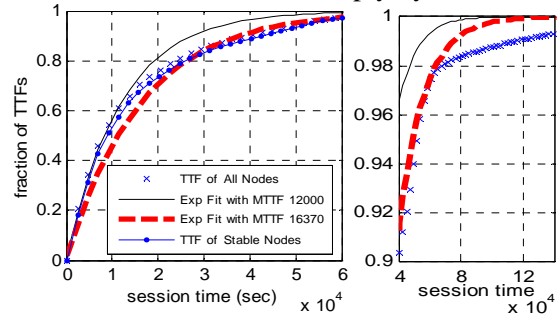
Session time (also called time to failure, TTF), which is defined as a continuous online period, is another important factor in the evaluation of short-term data durability [20]. We address the same question as Yalagandula did in non-P2P measurement [14]: does good availability always imply good mean session time (also called mean time to failure, MTTF) ?

We randomly select 2K peers both appearing in the first and last 5 days of March and plot the relationship of availability to MTTF in **Fig2**. This stable peer set eliminates peers with very short lifetime, which have very low availability in 31 days period. In fact, **Fig2** shows that better MTTF between two peers cannot be assumed through better availability alone. This problem will be even more evident when peer availability is over 0.6 and the mean session time appears evenly distributed in a range. Consequently, we should monitor and use availability and TTF respectively.

Another feature in **Fig2** is that MTTFs are virtually greater than 12 hours (43,200 seconds) when availabilities are over 0.7. We suspect these peers do not have a diurnal online pattern [9, 11], as will be validated in subsection 4.1.5.

The stable peers used in **Fig2** have much better

availabilities than all peers in March as shown in **Fig1**. However, we find their TTF distribution is very close to all peers' as plotted in **Fig3**. This implies that one can hardly distinguish the peers with shorter lifetime from others simply by session time.



**Fig3.** All session time distribution

#### 4.1.3. TTF of Peers and Exponential Fit

Only MTTF is insufficient, and we further need the TTF distribution to estimate the session time used in many analyses [7, 18-20].

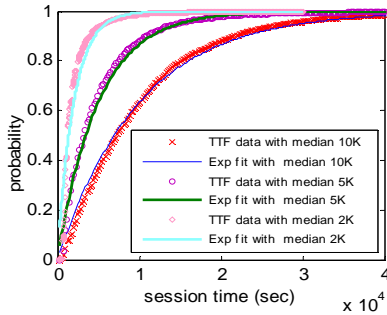
MTTF of all 3,978,163 sessions in March is 16,370 seconds. The TTF distribution of all peers and its exponential fit with an MTTF of 16,370 seconds are shown in **Fig3**. Though the exponential distribution roughly fits TTF distribution, it overestimates session length at first, and then underestimates session length. The zoomed-in part of **Fig3** indicates that TTF distribution may be a long-tailed distribution, so a peer already lived for a long time is less likely to leave the system than peer lived for a short while. Though long-tailed, we can use a pessimistic exponential distribution in analysis to ensure an underestimation over a large range, e.g. the exponential fit with an MTTF of 12,000 in **Fig3**.

Furthermore, the TTF distribution indicates that half an hour may be enough for the monitoring interval of the peer online detector in real system because over 80% sessions are longer than that.

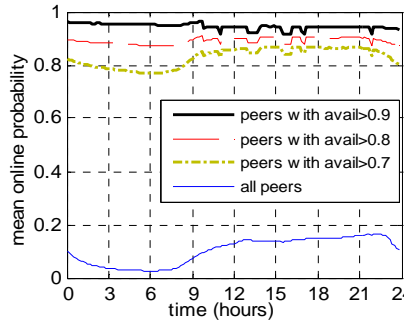
#### 4.1.4. Single Peer TTF and Exponential Fit

The TTFs of all peers do affect the durability of data replicated on an arbitrary set of peers, e.g. data stored in DHT-based storage systems [6]. However, the durability of data replicated on a selected set of peers, e.g. data in a directory-based storage system [6], is affected by the TTF of each peer in the set.

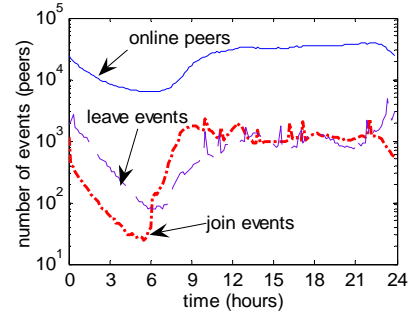
We assume that TTF distributions of peers with close MTTF are same and independent. Independence of online behavior was shown in [11]. Thus, we use the distribution of all TTFs of all peers with close MTTF to represent TTF distributions of these individual peers. We plot TTF distributions of 66 peers with MTTF in (1950, 2050), 188 peers with MTTF in (4950, 5050), 273 peers with MTTF in (9950, 10050) and their exponential fits respec-



**Fig4.** TTF distribution of a single node



**Fig5.** Diurnal online patterns



**Fig6.** Join and leave frequency

tively in **Fig4**<sup>1</sup>. We find that exponential distribution well fits the real distribution, which implies that the peer TTF is memoryless, i.e. the residual session length is independent with how long the peer has been online. We conduct a similar study on the distribution of TTR (time to repair) of individual peers, which also follows an exponential distribution.

#### 4.1.5. Diurnal Online Pattern

Previous studies [9, 11] have shown the diurnal pattern of peers online behaviors, so it seems almost infeasible to provide the same data availability at night-time as in day-time if only distributing data on peers within a few adjacent time zones. However, we may not make use of low available peers in practice because replicas stored on very low available peers contribute little to data availability, but consume no less bandwidth than those of replicas on high available peers. Therefore, we concentrate on online behavior of high available peers instead of all peers in this subsection.

The users in Maze are almost CERNET users within China, so they all can be considered in the same time zone as our log server. **Fig5** illustrates the online probability of 585 peers with availability over 0.9, 1300 peers with availability over 0.8, 2212 peers with availability over 0.7 and all peers in March. Though all peers' online behavior shows strong diurnal pattern, online probability of peers with availability over 0.7 does not vary much as a function of time. Thus, if the designers only use the high available peers in storage systems, they can make their designs regardless of the time-of-day effects. An interesting thing to note in **Fig5** is that peers with availability over 0.9 have less availability in hot online time areas for others from 12:00am to 23:00pm. This may because they are experienced users, and shut down Maze software sometimes when they are working.

#### 4.1.6. Join and Leave Frequency

We further characterize the join and leave fre-

quency of peers from a whole system view. **Fig6** plots the average number of join and leave events every 10 minutes and the average number of online peers as a function of time in March. The trace shows an average frequency of 0.75 joins (and leaves) per second in Maze system. Alternatively speaking, every 8 hours there will be roughly  $N$  joins (and leaves) in the system with  $N$  online peers. In such a dynamic environment, the multicast based failure detection such as [24] will be too costly, as will be storage systems with eager repair strategy [5] of bandwidth resources.

## 4.2. Long-term Dynamic

### 4.2.1. Permanent Departure and Detection

As we have seen the join and leave events are very frequent, it is almost infeasible to repair the stored data (or state) in the face of every peer leave. Though we can react only in the face of permanent peer departure from the system, it is difficult to distinguish departure from transient leave. In this subsection, we first investigate the departure rate, and then show how to detect permanent departures.

We define active peers at time  $t$  as the peers registered before  $t$  and appeared at least once in the following month after  $t$ . Thus, we get a population of 102,734 active peers on 1<sup>st</sup> March and 101,646 active peers on 4<sup>th</sup> April, which shows a steady-state system population.

According to above definition, peers that never appear within one month are considered departed from system. **Fig7** depicts the cumulative number of departures along time in March. This figure shows a constant turnover rate of about 4.3K departures per day. Consider a P2P storage system with a total size of  $S$  redundant data evenly stored on all  $N$  active peers, so each peer contributes  $S/N$  storage. Applying dynamic parameters in Maze, the total maintenance bandwidth for repairing lost data per day is  $S/102734 \times 4300 = S/23.9$ . Comparing with the model in [3], we get an average peer lifetime of about 24 days. According to the maintenance bandwidth analysis in [3], peers in Maze can support a storage of 50TB unique data even when the redundancy factor is 20.

<sup>1</sup> Nodes are from the stable node set used in section 4.1.2. This node set is used to eliminate nodes with only one session in the period, whose TTF distribution is meaningless.

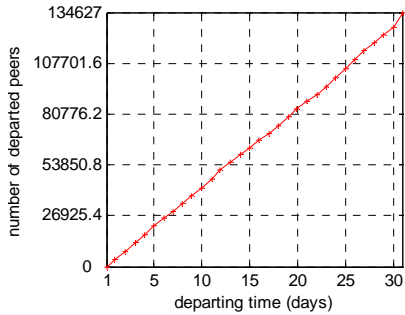


Fig7. Peer turnover rate

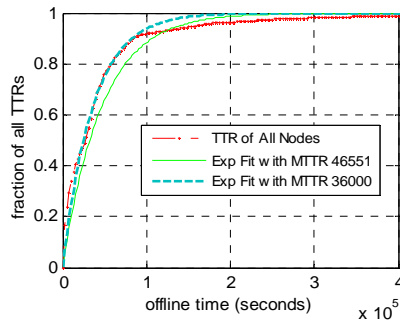


Fig8. TTR distribution

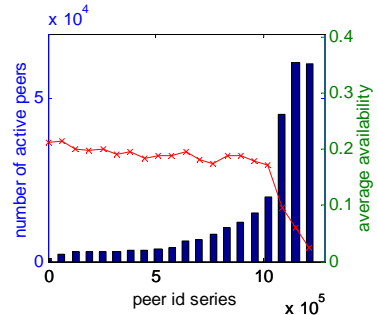


Fig9. Evolution of dynamic

The analytical large amount of storage the system can support relies on correct and immediate detection of peer departure. Nevertheless, the detection is quite hard in practice. If we use one month as threshold of the detector, we will get a false negative of about  $10^5$  peers. Though a smaller threshold can reduce false negative, it results in larger false positive. The false positive triggers repair process, so wastes system bandwidth for transient failures. Extra redundancy [5, 6, 21] is discussed to mask the false positive effects in storage system. The probability of false positive is a critical parameter in evaluating the overall maintenance bandwidth and the factor of extra redundancy. Then, the problem is how to estimate the probability of false positive.

Consider an offline interval  $T$  to be the detection threshold. Then, any transient leave with an offline time over  $T$  will be regarded as a departure, which is a false positive. Consequently, for this simple detector, false positive probability is determined by detection threshold and offline time distribution of transient leaves. We use TTR to refer offline time of transient leave in the following discussion. Fig8 plots the distribution of all  $3.7 \cdot 10^6$  TTRs and its exponential fits in March. This figure shows a similar long-tailed distribution with Fig3.

With the turnover rate and TTR distribution, we can evaluate the false negative and false positive of detectors with different detection thresholds.

#### 4.2.2. Dynamic Evolution and New Registered Peers Effect

As a long-running system, the dynamic of aged peers may differ a lot from newly registered peers, so we first give some insight into the long-term evolution of peer dynamic in this subsection, and then discuss how to eliminate the high dynamic effect caused by newly registered peers.

In Maze, each peer is assigned a sequential generated ID at registration. Thus, the ID larger than any ID in the logs before time  $t$  must be registered later than  $t$ . This provides us an opportunity to study the dynamic evolution of peers.

Till May 31<sup>st</sup>, the largest registered ID is 1,274,797. We evenly divide the ID space from 1 to 1,274,797 into twenty subspaces. In Fig9, we plot

the number of living peers and average availability of each subspace in May. The figure shows a steadily increase of the number of living peers along peer ID series, but a sharp increase in the latest registered ID subspaces. Contrarily, the steady decrease of availability dives dramatically in the latest ID subspaces. This figure indicates: first, elder peers are more stable than younger peers, i.e. elder peers are less likely to depart and have higher availability than younger peers; secondly, the latest registered peers are extraordinarily unstable.

We further conduct a fine-grained measurement of turnover rate of the latest registered peers. Table1 lists the numbers of peers found in logs of March, April and May, and peers are clustered into groups by their registration time. From the table, we find that only a fraction of near 1/3 peers registered in March are still alive in April, and so do the peers registered in April. The implication of the significant turnover rate of latest registered peers is that, we can greatly reduce the overall turnover rate by eliminating the high dynamic effect of latest registered peers.

	Mar	Apr	May
all IDs	236,273	214,493	276,162
IDs before Mar	102,734	59,263	45,778
new IDs in Mar	133,539	42,383	25,476
new IDs in Apr		108,197	39,045
new IDs in May			155,641

Table1. Evolution of living peers

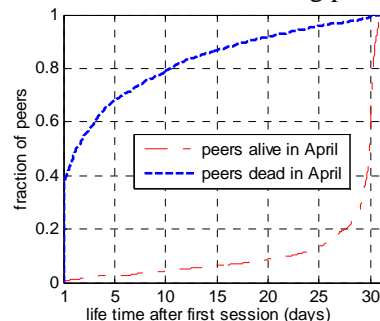


Fig10. Lifetime after first session in one month log

We use *stable peer* to refer a peer who has a lifetime longer than one month in this subsection. We note that the latest registered peer set contributes a large number of stable peers to the system, besides

a large fraction of high dynamic peers. Then, it is necessary to detect stable peers among all latest registered peers effectively and efficiently, so that we can make use of the newly registered peers as soon as possible.

We design a straightforward detector as follows. All newly registered peers are pessimistically regarded as unstable peers, and if a peer is still alive after a threshold of time  $T$  from the end of its first session, it is regarded as a stable peer. To study how the threshold affects false positive and false negative of this detector, we pick out all stable and unstable peers registered in March 1<sup>st</sup>, and illustrate their cumulative lifetime distributions in **Fig10**. **Fig10** shows that a fraction of 40% unstable peers never appear after first session, and unstable peers have much shorter lifetimes than stable peers. Assuming a detection threshold of 15 days, only about 10% unstable peers will be ignored by the detector, and only about 6% stable peers will be falsely reported as unstable ones. So this simple detector may work well in practice.

## 5. Summary

Based on system log, we measured the dynamic of the P2P system, Maze, and discussed the implications on system designs. Compared with previous measurements, we brought forth a series of new metrics required by recent analysis works. Our measurements and implications include: (1) crawler based methodology makes the previous measurements dramatically underestimate system dynamic; (2) good MTTF does not always imply good availability; (3) all peers' TTF distribution is long-tailed, but exponential distribution well fits a single peer's TTF; (4) high available peers do not have diurnal online patterns; (5) join and leave is very frequent; (6) turnover rate is not trivial, and we can estimate false positive and false negative of a departure detector; (7) aged peers are more stable than young peers, we can use a simple detector to eliminate the high dynamic effect of latest registered peers.

## Reference

1. Haeberlen, A., et al., *Fallacies in evaluating decentralized systems*. Proc. of the 5th International Workshop on Peer-to-Peer Systems, 2006.
2. Yang, M., et al., *Deployment of a large scale peer-to-peer social network*. In Proc. of 1st Workshop on Real, Large Distributed Systems, 2004.
3. Blake, C. and R. Rodrigues, *High Availability, Scalable Storage, Dynamic Peer Networks: Pick Two*. In 9th Workshop on Hot Topics in Operating Systems, 2003.
4. Rodrigues, R. and B. Liskov, *High Availability in DHTs: Erasure Coding vs. Replication*. Proc. of the 4th International Workshop on Peer-to-Peer Systems, 2005.
5. Bhagwan, R., et al., *Total Recall: System Support for Automated Availability Management*. In Proc. of the First ACM/Usenix Symposium on Networked Systems Design and Implementation (NSDI), 2004.
6. Weatherspoon, H., et al., *Long-Term Data Maintenance in Wide-Area Storage Systems: A Quantitative Approach*. Computer, 2005.
7. Chun, B., et al., *Efficient replica maintenance for distributed storage systems*. Proc. of the 3rd Symposium on Networked Systems Design and Implementation, 2006.
8. upstore, <http://upstore.grids.cn>. 2006.
9. Saroiu, S., P. Gummadi, and S. Gribble, *A measurement study of peer-to-peer file sharing systems*. Proceedings of Multimedia Computing and Networking (MMCN'02), 2002.
10. Sen, S. and J. Wang, *Analyzing Peer-To-Peer Traffic Across Large Networks*. IEEE/ACM TRANSACTIONS ON NETWORKING, 2004. 12(2): p. 219.
11. Bhagwan, R., S. Savage, and G. Voelker, *Understanding availability*. Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS'03), 2003.
12. Qiao, Y. and F.E. Bustamante, *Structured and Unstructured Overlays Under the Microscope - A Measurement-based View of Two P2P Systems That People Use*. In Proc. of the USENIX Annual Technical Conference, 2006.
13. Guha, S., N. Daswani, and R. Jain, *An Experimental Study of the Skype Peer-to-Peer VoIP System*. Proceedings of IPTPS, 2006.
14. Yalagandula, P., et al., *Beyond Availability: Towards a Deeper Understanding of Machine Failure Characteristics in Large Distributed Systems*. In Proc. of USENIX Workshop on Real, Large Distributed Systems (WORLDS), 2004.
15. Bolosky, W., et al., *Feasibility of a serverless distributed file system deployed on an existing set of desktop PCs*. Proceedings of the 2000 ACM SIGMETRICS international conference on Measurement and modeling of computer systems, 2000: p. 34-43.
16. Nurmi, D., J. Brevik, and R. Wolski, *Modeling Machine Availability in Enterprise and Wide-area Distributed Computing Environments*. In Proc. of Euro-Par, 2005.
17. Weatherspoon, H. and J. Kubiatowicz, *Erasure coding vs. replication: A quantitative comparison*. Proc. of IPTPS, 2002.
18. Utard, G. and A. Vernois, *Data durability in peer to peer storage systems*. Cluster Computing and the Grid, 2004. CCGrid 2004. IEEE International Symposium on, 2004
19. Ramabhadran, S. and J. Pasquale, *Analysis of long-running replicated systems*. Proc. of the 25th IEEE Annual Conference on Computer Communications (INFOCOM), 2006.
20. Tian, J., Y. Dai, and H. Wang, *Understanding Session Durability in Peer-to-Peer Storage System*. In Proc. of ICCS, 2006.
21. Tati, K. and G. Voelker, *On object maintenance in peer-to-peer systems*. Proc. of the 5th International Workshop on Peer-to-Peer Systems, 2006.
22. Yang, M., et al., *An Empirical Study of Free-Riding Behavior in the Maze P2P File-Sharing System*. In 4th International Workshop on Peer-to-Peer Systems, 2005.
23. Yang, M., Y. Dai, and J. Tian, *Analyzing peer-to-peer Traffic's Impact on Large Scale Networks*. In Proc of ICCS, 2006.
24. Zhang, Z., et al., *BitVault: a Highly Reliable Distributed Data Retention Platform*. under submission, 2005.