Characterizing Files in the Modern Gnutella Network: A Measurement Study

Shanyu Zhao, Daniel Stutzbach, Reza Rejaie
Computer and Information Science Department
University of Oregon
{szhao, agthorr, reza}@cs.uoregon.edu

The Internet has witnessed an explosive increase in popularity of Peer-to-Peer (P2P) file-sharing applications during the past few years. As these applications becomes more popular, it becomes increasingly more important to characterize their behavior in order to identify their performance bottlenecks and limitations as well as their impact on the network.

In this paper, we present a measurement study on characteristics of available files in the modern Gnutella system. We developed a new methodology to capture accurate "snapshots" of available files in a large scale P2P system. This methodology was implemented in a parallel crawler that captures the *entire* overlay topology of the system where each peer in the overlay is annotated with its available files. We have captured tens of snapshots of Gnutella system and conducted three types of analysis on available files: *(i)* Static analysis, *(ii)* Topological/Geographical analysis and *(iii)* Dynamic analysis. Our results reveal several interesting properties of available files in Gnutella that can be leveraged to improve design and evaluations of P2P file-sharing applications.

Keywords

Network Measurement, Peer-to-Peer, File Sharing, Distribution of Content, Gnutella

I. Introduction

During the past few years, the Internet has witnessed an explosive increase in the popularity of Peer-to-Peer (P2P) file sharing applications, which are primarily used for exchanging multimedia files. Today's popular P2P file-sharing applications such as eDonkey, FastTrack, and Gnutella have more than one million users each at any point of time [1], and make up a significant fraction of network traffic [2]. In these file-sharing applications, each peer offers a subset of its files to the system and participating peers collectively form an overlay used to search for files among those available throughout the system.

As file sharing applications become more popular, characterizing their behavior becomes more important because it reveals how well these systems perform in practice, whether they can be further improved, and their impact on the network. To fully characterize the behavior of file-sharing applications, three equally important and related aspects of these applications should be examined through measurement: *(i)* Overlay topology [3, 4], *(ii)* Query workload [5], and *(iii)* Available files [6]. In particular, characterizing available files among participating peers is valuable for several reasons. First, it reveals the properties, distribution and heterogeneity of the resources contributed (*i.e.,* storage space and available files) by users of the system. Second, it allows us to identify any potential design anomaly that might be exposed in a practical setting or any opportunity that can be used to improve performance of these systems. Third, collected traces and derived characteristics of available files through measurement can be also used to conduct more realistic simulations or analytical modeling on available files in P2P systems.

During the past few years, a handful of previous studies have characterized the distribution of shared files in various P2P file sharing systems [6–9]. While these studies shed an insightful light on the characteristics of files in file-sharing applications, they have several limitations. First, almost all the previous studies have focused on a small population of peers in file-sharing systems (*i.e.,* less than 20k peers). However, to our knowledge, none of these studies have verified whether the derived characteristics of files from the subset of captured peers indeed represent the behavior of the entire population. Second, many of the previous studies (except [6, 7]) are more than three years old and thus rather outdated. During the past few years, P2P file-sharing applications have significantly grown in size and have incorporated new features. In particular, the top three popular file sharing applications have adopted a two-tier architecture to improve their scalability. However, the effect of this new architecture on the characteristics of files has not been studied. Third, previous studies have only characterized various aspects of available files among peers in a given snapshot. Therefore, any possible impact of the overlay topology on file distribution nor dynamics of file characteristics over time has been examined in earlier studies.

In this paper, we empirically characterize available files across the *entire* population of peers (more than 1 million) in the modern Gnutella network. We develop a new measurement methodology that allows us *(i)* to properly identify a group of peers that present an accurate "snapshot" of the system at a particular point of time, and *(ii)* to capture both available files at each peer and pair-wise connectivity among peers (*i.e.,* overlay topology) for a given snapshot. We have developed a fast parallel crawler called Cruiser [10]. Using Cruiser, we have captured more than 30 snapshots of the files available in Gnutella with more than 100 million distinct files in each snapshot. Using these snapshots, we conduct the following analysis:

- **Static Analysis:** We examine properties of contributed resources (*i.e.,* files, and storage space) by participating peers in a single snapshot of the system.

- **Topological/Geographical Analysis:** We investigate whether the pattern of file distribution among peers has any correlation with the overlay topology or with the geographic location of the peer.
- **Dynamic Analysis:** We study how the popularity of available files changes over different timescales.

Our main findings can be summarized as follows:

- Free riding has significantly decreased among Gnutella users during the past few years and is significantly lower than other P2P file-sharing applications such as eDonkey. While the ratio of free riders among Ultrapeers is slightly lower than among leaf peers, there is no correlation between a peers' uptime and their tendency to free ride.
- The number of shared files and contributed storage space by individual peers both follow a power-law distribution. Compared to earlier studies, Gnutella users contribute significantly more disk space but share approximately the same number of files.
- The popularity distribution of individual files follows a Zipf distribution which means that a small number of files are extremely popular.
- The most popular file type is the MP3 file, which accounting for two-thirds of all files and one-third of all bytes. Both the popularity and occupied space by video files has tripled over the past few years. However, the number of video files are less than one-tenth of audio files but they occupy 25% more bytes. 93% of bytes in the system are occupied by multimedia files.
- Files are rather randomly distributed throughout the overlay and there is no strong correlation between the files shared by neighboring peers in the overlay topology. However, files shared by geographically co-located peers have a visible degree of similarity.
- Shared files by individual peers slowly change over the timescale of days. However, over the entire system, more popular files experience larger variations in their popularity. Furthermore, the recent past trend in variations of a file popularity seems to predict its changes in popularity in the near future.

A. Why Characterizing Gnutella?

We decided to conduct our empirical study on Gnutella based on a number of considerations. First, Gnutella is one of the top three most popular P2P file-sharing networks on the Internet [1]. During the past year the population of concurrent Gnutella users has tripled and is currently around 2 million. Therefore, while Gnutella is not the most popular, but it is definitely a large scale and representative file-sharing applications with an active user population.

Second, Gnutella has a protocol hook that allows a list of shared files to be extracted from a peer. This eliminates the need for kludgey techniques that might introduce significant error.

Finally, Gnutella is one of the most studied P2P systems in the literature. This enables us to compare and contrast the behavior of modern Gnutella with earlier empirical studies on Gnutella and gain insights on changes in the system.

The rest of this paper is organized as follows: Section II presents an overview of previous studies. Section IV discusses the challenges in capturing accurate snapshots and describes our measurement methodology and tools. Section V, VI and VII present static analysis, topological/geographic analysis, and dynamics analysis of files in the Gnutella network, respectively. Section VIII concludes the paper and sketches our future plans.

II. Related Work

Many measurement studies have reported on different properties of P2P file-sharing networks including: *(i)* churn [11,12], *(ii)* overlay topology structure [3,4,13,14], *(iii)* query traffic [5], *(iv)* data traffic [15–17], and *(v)* files shared [6,7]. The topic of this paper is the later: characterizing the files shared by users. We are aware of only two other studies that focus on the files shared by users.

First, Fessant *et al.* [6] examined correlations between files, using data collected from 12,000 eDonkey clients over a three day period in 2003. They showed that the popularity of files stored in file-sharing systems is heavily skewed, following a Zipf distribution. Also, users have noticeable interests, with 30% of files having a correlation of at least 60% with at least one other file. When two peers have 10 files in common, there's an 80% chance they have at least one more file in common. The probability is close to 100% if they have at least 50 files in common.

Second, Liang *et al.* [7] recently analyzed the nature and magnitude of deliberately corrupted files ("pollution") in Kazaa. To combat P2P sharing of copyrighted content, some companies intentionally inject decoy files, which have the same file name as a popular song. They developed a multi-threaded crawler that queries all 30,000 Kazaa super-nodes for seven popular songs over the course of one hour. They show that the popularity of different versions of a song also follows a Zipf distribution. For most of the seven popular songs, over 50% of the copies are polluted.

A few other studies have examined the files shared by users as part of broader measurement studies on peer-to-peer systems. Examining 5,000 Gnutella peers per week in 2001, Chu, Labonte, and Levine [8] examined peer churn, the distribution of file popularity, and the distribution of transfer popularity. They found that file popularity follows a log-quadratic distribution (which can be thought of as a second-order Zipf distribution). Also in 2001, Saroiu, Gummadi, and Gribble [9] examined many characteristics of peers in Napster and Gnutella, such as their bottleneck bandwidth, latency, uptime, and number of shared files. They found that the number of shared files was heavily skewed.

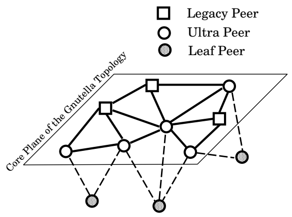


Fig. 1. Two-Tier Topology of Modern Gnutella

While our work focuses on the files shared by users in P2P systems, other studies [15, 16, 18] have focused on the files being actively transferred. These two properties are correlated in an obvious way, so their work is closely related to, and complements, ours. The transfer of files contributes to the pool of files being shared, but is subject to shorter-term trends. In contrast, the files shared (*i.e.,* made available) by a user may be the result of transfers over the course of months or years, followed by a gradual pruning of unwanted files.

Data traffic studies capture packets at a router located between some user population and the rest of the Internet. Gummadi *et al.* [15] analyzed a 200-day trace of Kazaa traffic collected at the University of Washington, demonstrating that file transfers in Kazaa do not quite follow a Zipf distribution and that this difference is due to the "fetch-at-most-once" nature of downloads in file-sharing applications. Another analysis of Kazaa traffic was conducted by Leibowitz, Ripeanu, and Wierzbiciki [16] at a large Israeli ISP. They examined the changing popularity of files being transferred, which is similar to our analysis of the changing popularity of files being stored, presented in Section VII. They also showed that small-world properties exist among data-sharing relationships.

Our studies differs from the few previous studies on the files shared by users. First, the population of peers used in study is significantly larger. In total, we examined a complete snapshot of a the Gnutella network with around 2 million distinct peers in each snapshot. Second, we explore two properties of the files which have not previously been studied: *(i)* the relationship between the files shared by users and the overlay topology structure, and *(ii)* the change in popularity of files stored over time.

III. Overview of Modern Gnutella

Gnutella is widely regarded as the first fully decentralized peer-to-peer file-sharing system. However, it has evolved considerably since its initial release in early 2000, and grown dramatically in size over the last year [3]. Today, Gnutella is one of the largest P2P networks in operation [1].

Similar to many unstructured P2P networks, each Gnutella peer joins the network by establishing TCP connections to several existing peers. In the original Gnutella protocol, participating peers formed an overlay and use TTL-scoped flooding to search other peers. It was not long before the limited scalability of this simple flooding scheme became apparent. To improve the scalability of the Gnutella protocol, most modern Gnutella clients adopt a two-tier overlay structure along with a dynamic query distribution mechanism.

Two aspects of Gnutella are pertinent to our study. First, because one of our goals is to examine correlations between the distribution of shared files and location in the overlay topology, a general understanding of Gnutella's structure is required. In our prior work, we conducted a detailed study of the Gnutella overlay topology [3, 13]. Second, we make use of Gnutella's Browse-Host extension to acquire the list of files shared by each peer.

A. Two-Tier Topology

Modern Gnutella clients implement a two-tiered overlay structure by dividing peers into two groups: *ultrapeers* (or superpeers) and *leaf peers*. Each ultrapeer neighbors with several other ultrapeers within the top-level overlay. The majority of the peers are leaves that are connected to the overlay through a few (2 to 3) ultrapeers. High-bandwidth, unfirewalled leaf peers become ultrapeers on demand in order to maintain a proper ultrapeer-to-leaf ratio. Those few peers that do not implement the ultrapeer feature can only reside in the top-level overlay and do not accept any leaves. We refer to these peers as legacy peers. When a leaf connects to an ultrapeer, it uploads a set of hashes of its filename keywords to that ultrapeer. This allows the ultrapeer to only forward messages to the leaves that might have matching files. Leaf peers never forward messages. This approach reduces the number of messages forwarded towards leaf peers which in turn increases the scalability of the network by a constant factor.

B. The Browse-Host Extension

One important reason that we choose modern Gnutella for file characterization is because Gnutella has a suite of open and moderately well-documented protocols. The Browse-Host extension [19] is a Gnutella protocol extension designed to allow one peer to view the list of files shared (called a *sharing list*) by another peer. It's intended use is to allow users with similar interests to learn about new material which may appeal to them.

Browse-Host is supported by the two major Gnutella implementations, BearShare and LimeWire, among others. These two implementations combined compose roughly 95% of Gnutella ultrapeers [3], giving us a reasonable expectation that we may use Browse-Host to study the files shared by most peers in Gnutella.

IV. Measurement Methodology

Our goal is to capture a *snapshot* of available files in the Gnutella network which contains all participating peers in the network, the available files at each peer, and the pair-wise connectivity among participating peers (*i.e.,* the overlay topology) at a given point of time. A common approach to capture a snapshot is to deploy a P2P crawler. Given a set of initial peers, a crawler contacts individual peers to capture their available files and collect information about new peers in the session. Thus, the crawler progressively learns about more peers in the session and contacts them until no other new peers are available. However, because of the dynamics of peer participation (or churn) and the slow speed of crawlers in practice, captured snapshot by a crawler are inherently *distorted* [20]. More specifically, as the crawler explores the overlay, many peers join or leave the system and change the overlay topology or may change the set of files they are sharing. Therefore, the captured snapshot contains a combination of peers that were present during a crawl. This problem is further aggravated in large overlays since a sufficiently large number of new peers may significantly increase the duration of a crawl and thus inflate the population of peers in a snapshot[1].

Previous studies implicitly addressed this problem by adopting one of the following sampling schemes to capture a partial snapshot of a P2P system: *(i) Partial Snapshot Through a Short Crawl*: Some studies [9] captures a small group of peers (*i.e.,* a partial snapshot) through a short crawl and assumed that the captured peers properly represent the entire population. *(ii) Periodic Probe of a Fixed Group*: Other studies obtained information about some participating peers and periodically probe the same group of peers and collect information about their available files [8]. In the absence of any solid understanding of file characteristics in P2P system[2], it is not clear whether these sampling strategies capture a representative population of peers.

We developed the following measurement methodology to capture a representative snapshot of Gnutella network. We try to capture the entire population of participating peers in the Gnutella network (*i.e.,* a complete snapshot) to minimize any potential bias in our characterization. To improve the accuracy of captured snapshots, they are collected in three steps. First, we conduct a *topology crawl* to quickly capture all participating peers and their pair-wise connectivity, *i.e.,* capturing the overlay topology. Then, we conduct a *content crawl* and collect the list of files available at each one of the peers identified during the topology crawl. Once the content crawl is completed, we initiate another topology crawl in order to distinguish short-lived and long-lived peers in the initial topology crawl. More specifically, any peer that is present in both topology crawls is considered long-lived while other peers that have left the system between the two topology crawls and are short-lived. This approach creates a snapshot of the overlay topology where each node is annotated with its available file and a label that determines whether it is long- or short-lived.

To justify our methodology, we note that the time required to obtain the list of available files is significantly longer than that for neighbor information. For example, the time to obtain a list of neighbor peers from a peer may take less than a second whereas the list of available files may take a few minutes to download. This implies that a topology crawl is much shorter than a content crawl. Therefore, decoupling of topology and content crawl enables us to capture a much more accurate snapshot of the entire population of peers during a short period. Since some of the captured peers in the first topology crawl have left the system during the content crawl, the collected content in our measurement is slightly biased towards peers with longer uptime.

Cruiser, A Parallel Crawler: We have developed a parallel P2P crawler, called Cruiser [10], that can crawl an overlay orders of magnitude faster than any previous crawler. Cruiser achieve this goal by significantly increasing the degree of concurrency of the crawling process. Toward this end, Cruiser adopts a master-slave architecture where each slave crawls hundreds of peers simultaneously and the master coordinates among multiple slaves. This architecture allows us to run Cruiser on multiple co-located or distributed boxes to further increase the crawling speed. Using six off-the-shelf 1 GHz GNU/Linux boxes in our lab, Cruiser can perform a topology crawl for more than two million Gnutella peers in less than 15 minutes, and perform a content crawl within 5.5 hours, *i.e.,* capturing the annotated snapshot takes 6 hours, (5.5hr + 15min + 15min). During the content crawl, Cruiser collects the file name and content hash (SHA1) for each shared file on every reachable peer and generates a 10GB log file.

Dataset: During June of 2005, we have captured more than 30 snapshots of the Gnutella network annotated with the list of files available at each peer and their type as long/short-lived. Table I summarizes the statistics of several snapshots. As shown in this table, we divide captured peers into three groups: Ultrapeers, Leaf peers, and Legacy peers. By legacy peers, we are referring to those few Gnutella clients running older version of the software that do not support the two-tier architecture; these peers remain in the top-level overlay. In each group, a subset of peers might be unreachable by our crawlers for one of the following reasons: *(i)* firewalls or NAT blocking incoming traffic, *(ii)* severe network or processor congestion at the peer, or *(iii)* the peer has departed. Since ultrapeers are not allowed to be firewalled, any reported connection error for ultrapeers

[1] In the extreme case, the crawler may never terminate since there are always new peers to contact

[2] Existing empirical studies used one of the above strategies, therefore it is not clear to what extent their findings are reliable

Crawl Date	Ultrapeers				Leaf Peers				Legacy Peers				Total Peers			
	Number	% C.E.	% T.O.	% C.L.	Number	% C.E.	% T.O.	% C.L.	Number	% C.E.	% T.O.	% C.L.	Number	% C.E.	% T.O.	% C.L.
June 11	279,512	30.96	3.77	3.42	4,168	54.10	3.14	6.91	1,855,271	84.00	1.01	0.95	2,138,951	77.01	1.37	1.29
June 12	287,071	33.19	3.66	3.38	4,338	58.88	2.95	7.42	1,893,212	84.55	1.02	0.94	2,184,621	77.75	1.37	1.28
June 13	281,472	34.90	3.48	2.97	4,467	59.12	2.60	6.92	1,932,944	85.39	0.90	0.88	2,218,883	78.93	1.23	1.15
June 14	287,147	36.76	3.20	2.63	4,072	62.65	2.41	5.48	2,024,716	85.79	0.85	0.83	2,315,935	79.67	1.15	1.06
June 15	285,101	37.41	3.68	2.89	4,032	63.86	2.78	4.99	2,019,319	85.82	0.92	0.86	2,308,452	79.81	1.27	1.12

TABLE I

CRAWLING STATISTICS FOR SEVERAL SNAPSHOTS CAPTURED DURING JUNE, 2005, C.E.: CONNECTION ERROR, T.O.: TIMEOUT, C.L.: CONNECTION LOST

indicates that the contacted peer has departed. However, connection errors for leaf peers might occur due to peer departure or a firewall[3]. In our earlier study of Gnutella [3], we showed that about half of all leaf peers leave the overlay after a 5 hour period. Independent online statistics [21] report that around 70% of leaves in the Gnutella network are firewalled. These evidences support the accuracy of the high ratio of connection errors that we experienced for leaf peers. In summary, while our captured snapshots are rather complete, we can directly contact only 20% of all peers in one snapshot (around half a million peers) to obtain their list of available files.

A. *Other Challenges and Problems*

We briefly discuss several other problems that we experienced during our data collection and data processing that are worth sharing.

Long-lived TCP Connection: Although Cruiser has a timeout mechanism that closes any idle connections after 20 seconds, we noticed that some crawls do not complete after the crawling queue becomes empty. Further examinations revealed that around 80 peers in each crawl send their data at an extremely low speed (around 20 bytes per second) which prevents Cruiser from closing their connections. We instructed Cruiser to terminate a crawl a few minutes after its crawling queue becomes empty. Given the negligible number of these misbehaved peers, this should not have any affect on our analysis.

File Identity: We use the content hash of a file returned by the target peer to uniquely identify each file. In our initial measurements, we observed many files with the same names but different content hashes (*e.g.,* setup.exe, login.bmp). This illustrates that the trimmed (or even complete) file name that was used by previous studies [8], is not a reliable identifier. We discovered around 3,500 files without content hash value in each snapshot and eliminated them from our analysis.

Post-processing: To compute the popularity of individual files in the system, we needed to keep track of more than 100 million distinct files in the system which resulted in memory bottlenecks in our analysis. We leveraged the skewed distribution of popularity to address this problem as follows: We divide captured peers in a snapshot into seven segments and calculate the popularity of files within in each segment. Then, we trimmed all files that had less than 10 copies in a segment (which deletes several million distinct files!), and combined all the trimmed results for different segments. This restricts us from performing analysis on the least popular files (those with fewer than 70 copies in the entire network) but does not significantly affect analysis performed on more popular files.

V. STATIC ANALYSIS

In this section, we examine characteristics of available files across all peers in a single snapshot regardless of their location in the overlay topology. In particular, we examine the following issues: *(i)* the ratio of free riders, *(ii)* the degree of resource sharing among cooperative peers, *(iii)* the distribution of file popularity, and *(iv)* the distribution of file types. We compare our findings with previous studies to identify any potential changes in the behavior of P2P systems with respect to the above issues over the past few years. To allow cross-comparison of different results in this section, we focus on one of the snapshots listed in table I, was captured on June 13th, 2005. However, we have examined several other snapshots and observed similar behavior. Therefore, these results are representative behavior in our snapshots.

A. *Ratio of Free Riders*

The success of P2P file sharing system depends on the willingness of participating peers to share files. However, previous studies have frequently reported that participating peers do not have an incentive to contribute their resources, such as disk space and network bandwidth, to the system and thus only use resources offered by others, *i.e.,* become "free riders". In particular, Adar *et al.* [22] reported that 66% of Gnutella peers were free riders in 2000, while a study by Saroiu *et al.* [9] found 25% were free riders, with 75% of peers sharing less than 100 files in 2002. A recent study also reported 68% were free riders in eDonkey [6].

Table II presents the degree of free riding among Gnutella peers in our June 13 snapshot. We separated ultrapeers (first row) and leaf peers (second row) to examine any potential difference in free riding between them. We further divide ultrapeers (row 3 and 4) and leaf peers (row 5 and 6) into short-lived and long-lived based on their presence in the second topology crawl as we discuss in Section IV. We distinguish two types of uncooperative peers:

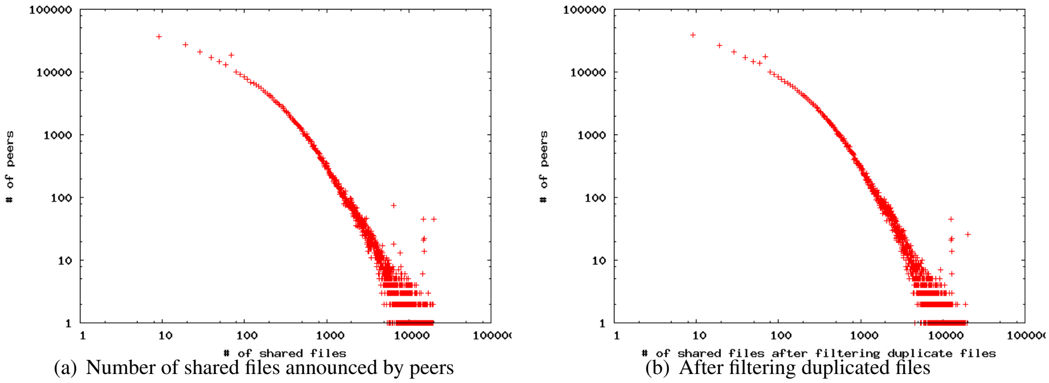
- *Not Reporting* that captures those peers whose Gnutella client is configured to not release a sharing list.

[3] We are not aware of any reliable technique to distinguish between these two scenarios.

	Number	Contacted	% Not Reporting	% Free Riders	Files/Peer
Ultra	281,472	165,083	3.69	12.11	352
Leaf	1,932,944	245,429	4.33	15.49	332
Long-lived Ultra	150,462	129,667	3.78	12.27	349
Short-lived Ultra	131,010	35,416	3.35	11.52	363
Long-lived Leaf	917,758	163,541	4.61	16.26	350
Short-lived Leaf	1,015,186	81,888	3.76	13.94	297
Total	2,214,416	410,512	4.24	15.06	334

TABLE II

FREE-RIDING AND SHARING STATISTICS OF THE SNAPSHOT ON JUNE 13TH, 2005



(a) Number of shared files announced by peers

(b) After filtering duplicated files

Fig. 2. Distribution of the number of shared files, June 13 snapshot

- *Free Riders* are those peers that do not share any files

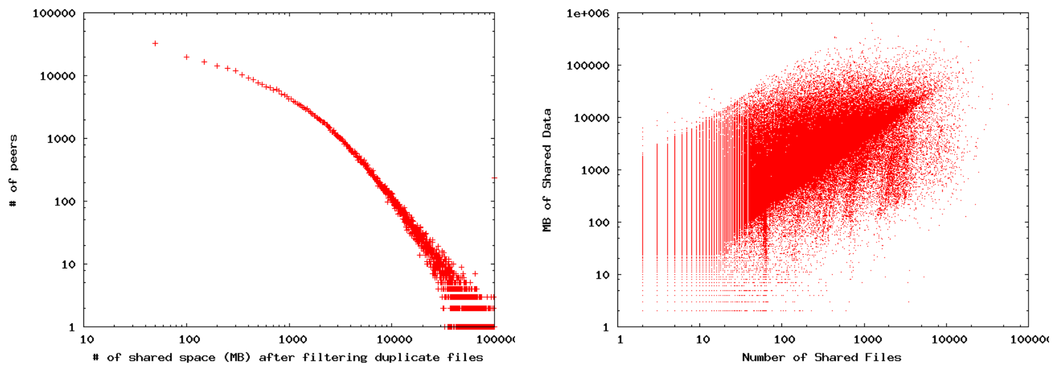
For each one of the above group, the corresponding row in table II presents the total population, the number of contacted peers, the ratio of "Not Reporting" peers, the ratio of free riders, and the average number of shared files over cooperative peers in that group (*i.e.,* excluding "Not Reporting" and free riders).

Table II shows several interesting points as follows: First, the ratio of free riders in Gnutella has significantly dropped from 25% in 2002 to 15% among all participating peers (*i.e.,* last row), and it is drastically lower than the 68% recently reported in eDonkey [6]. We speculate that the observed drop in the ratio of free riders is due to the increase in access link bandwidth for average Internet users and marketing efforts by the Gnutella vendors encouraging their users to share. Furthermore, ratio of free riding ultra peers (12.11%) is some what lower than that in leaf peers (15.49%). However, since leaf peers constitute a larger portion of the total population (87%), their behavior has a bigger impact on system performance. Second, Table II reveals that long-lived peers have a slightly higher ratio of free riders compare to short-lived peers. However, this effect is more visible among leaf peers. Third, the average number of shared files reveal that the user sharing behavior does not strongly correlate with their uptime. Although long-lived ultrapeers tend to share fewer files than short-lived ultrapeers, leaves exhibit the opposite tendency. Examination of other snapshots showed that the ratio of free riding was consistent during our measurement period.

B. *Degree of Resource Sharing Among Cooperative Peers*

We now turn our attention to cooperative peers and characterize their willingness to share their resources, both files and storage space, with other peers. Figure 2(a) plots the distribution of the number of peers that are willing to share x files in one snapshot. This distribution largely conforms to a power-law with a flat head. We noticed that the sharing lists of many peers contain duplicated files. This is because most Gnutella clients simply put various folders under the sharing folder which results in reporting the files in all those directories in their sharing list. As a result, any duplicate file in different shared folders are reported multiple times in their sharing list. Figure 2(b) shows the same distribution after removing duplicated files from each sharing list, *i.e.,* around 10,308,240 files or 8.75% among all files. The similarity between Figures 2(a) and 2(b) shows that duplicate files do not have an obvious impact on the distribution. Future design of Gnutella client software should remove duplicate files when constructing the sharing list to improve efficiency. In the remaining analysis in this paper, we exclude duplicate files.

To illustrate a different angle of resource sharing, Figure 3(a) depicts the distribution of shared disk space (in MByte) among cooperative peers in one snapshot after removing duplicate files. This figure also shows a power law distribution which means that most of the participating peers contribute small disk space ($<$ 100 MByte) while a little number of peers



(a) Distribution of the shared space among cooperative peers (b) Correlation between the number of shared files and shared space in MByte

Fig. 3. Aspects of file sizes

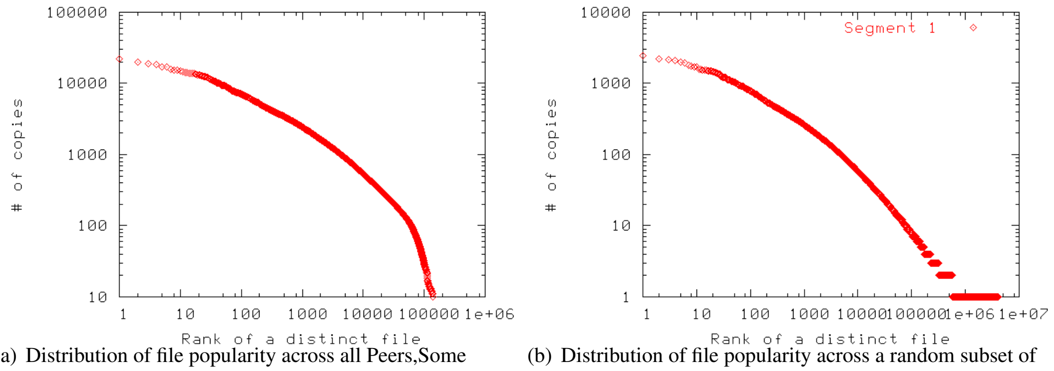
contribute a very large storage space (50–100 GByte).

Saroiu *et al.* [9] reported a strong correlation between the number of shared files and the volume of shared data for Gnutella peers in 2002. Figure 3(b) shows a similar scatter-plot for the number of shared files versus the shared disk space, across all cooperative peers in one snapshot, *i.e.,* one point per peer. This correlation is not as strong as that reported by Saroiu *et al.* three years ago. More specifically, peers sharing 1 to 1000 files in our snapshots exhibit two orders of magnitude wider variation in their contributed shared space compared with Gnutella peers three years ago. In a nutshell, current Gnutella users are using significantly more disk space but sharing a similar number of files. There is a discernable line with the slope around 3.7MByte/file in Figure 3(b) which is the typical size of a MP3 audio file.

C. *File Popularity Distribution*

The distribution of popularity of individual files throughout the system is an interesting property that shows the degree of similarities in available files at different peers. Previous studies have reported rather inconsistent results on the distribution of file popularity in various file-sharing applications. Chu *et al.* [8] showed that the file popularity follows a log-quadratic distribution, which can be viewed as a second-order Zipf distribution, among Gnutella peers in 2001. However, Fessant *et al.* [6] recently reported a Zipf distribution for the file popularity in eDonkey. Furthermore, none of these studies have captured a large number of peers.

Figure 4(a) shows the distribution of file popularity as a function of its rank (*i.e.,* popularity) for all peers in Gnutella. This distribution is across more than 104 million unique files reported by 0.4 million peers. This figure shows that file popularity mostly follows a Zipf distribution, except for its steep tail. The steep drop at the tail of the distribution is caused by under-counting the unpopular files (with fewer than 70 copies) that we deleted during the post-processing of each segment of each snapshot (see the discussion on post-processing in Section IV for more details). Figure 4(b) depicts the distribution of file popularity for a random subset of peers (only in one segment of the snapshot) which does not under-count unpopular peers. This figure clearly show that there are a huge number of unpopular files that only have one or two copies among peers in this



(a) Distribution of file popularity across all Peers,Some unpopular files were under-counted in our post-processing (b) Distribution of file popularity across a random subset of peers in one snapshot

Fig. 4. File Popularity Distribution

Type	% file number	% disk space
mp3	61.45	37.80
jpg	7.54	0.21
gif	3.14	0.003
wma	2.70	1.28
htm	2.69	0.004
exe	2.65	0.60
wmv	2.62	3.41
mpg	2.36	23.18
wav	1.86	0.71
txt	1.62	0.005
Subtotal	88.63	67.20

TABLE III

TOP 10 POPULAR FILE TYPES ON JUNE 13, 2005

Major Audio Types		
Type	**% file number**	**% disk space**
mp3	61.06-61.54	36.96-38.40
wma	2.69-2.76	1.28-1.36
wav	1.83-1.98	0.69-0.73
m4a	1.33-1.47	0.71-0.78
Subtotal	67.14-67.58	39.68-41.21
Major Video Types		
Type	**% file number**	**% disk space**
wmv	2.10-2.73	3.41-3.54
mpg	2.36-2.46	23.14-23.72
avi	0.79-0.81	24.04-25.10
asf	0.14-0.15	0.64-0.66
mov	0.07-0.08	0.16-0.18
rm	0.06-0.06	0.16-0.17
Subtotal	5.65-6.16	52.02-53.16

TABLE IV

MAJOR AUDIO AND VIDEO TYPES DURING JUNE 8 - JUNE 16, 2005

segment. In summary, the distribution of file popularity among Gnutella users follows Zipf's law. This means a small number of files are extremely popular while most of the files have very few copies among peers. Examination of other snapshots revealed that file popularity exhibits the same distribution (with slightly different slope) across different snapshots.

D. File Type Analysis

We have also examined the distribution of files available in Gnutella between different types of video and audio formats. Our results show what files types are mostly shared by Gnutella users. Chu *et al.* [8] conducted similar analysis for Gnutella peers in 2001 and reported that audio files constitute 67.2% of files and 79.2% of bytes but video files are significantly less popular and only contribute 2.1% of files and 19.1% of bytes.

Using our snapshots, we analyze the various type of audio and video files based on file extensions. In this snapshot, our crawler has successfully contacted 18.5% of all the peers in the topology and collected information for 104,340,947 files, about 813 terabytes in total. If we assume the unreachable peers have the same profiles, the total capacity of the Gnutella network at the moment is estimated to be about 4400 terabytes. Table III lists the top ten most popular file types (in terms of number of files), along with their popularity and their contribution in the disk space. This table shows that mp3 audio files are significantly more popular than any other file type and occupy more than one third of all disk space across the system. Although non-media files (*i.e.,* jpg, gif, htm, exe, txt) are among the top ten most popular types, audio and video files collectively occupy more than 93% of disk space in the system.

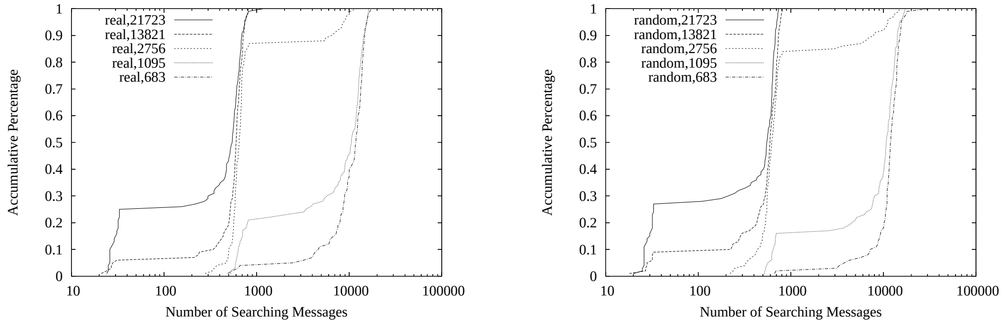
Table IV shows the range of popularity for the most popular audio and video file across various snapshots. The subtotal rows in this table clearly demonstrate that audio files account for 67% of files and 40% of bytes whereas video files constitute around 6% of files but 52.5% of bytes among Gnutella peers. Comparing to the reported results by Chu at al. earlier, video files have become three times more popular and occupy almost three times the fraction of space they once did.

VI. TOPOLOGICAL ANALYSIS

As reported in [6], the probability that any two clients having at least x files in common share at least one more file in common increases rapidly with x. This implies that the more similar the two peers' sharing lists are, the more likely that a new file request from one peer could be satisfied by the other peer. Thus, the similarity of two peers may be used as a good guidance for file searching, as shown by Sripanidkulchai *et al.* [18]. Their results show that files in Gnutella are *clustered*, which is undoubtedly the result of several contributing factors. In this section, we explore two factors which might contribute to the clustering observed in prior work.

One interesting quality of unstructured P2P systems is that they do not provide global search. When a user issues a query, their peer searches a group of nearby hosts in the overlay topology. Basically, this allows these systems to scale to any size. No matter the size of the system, each peer can search n peers around it. The efficiency of the system determines the size of the search radius.

One can see how this architecture might lead to some files being present only in certain areas of the network. When a file is first inserted into the network, only nearby peers will be able to find it. As other users copy the file, the accessibility of the file gradually grows outwards until there are enough copies in diverse enough locations that anyone can find the file.



(a) Real Annotation

(b) Random Annotation

Fig. 5. CDF of number of flooding messages needed to reach 5 copies of a particular file.

However, as peers arrive and depart, the overlay topology shifts and changes, leading to files "moving" from one part of the topology to another. The key question is: which is the dominating effect? Do files tend to be *topologically clustered* or are they evenly distributed throughout? Files could also be *geographically clustered*, with some files being significantly more popular in some parts of the world than in others.

A. Topological Clustering

To examine whether the availability of files is related to the topological structure, we use a snapshot of the overlay topology in conjunction with a snapshot of the files shared by the peers in that topology. Again, we use the topology and content snapshots taken on June 13, 2005 as an example, though we have performed this analysis for other snapshots as well.

We use a trace-driven approach, simulating an expanding-ring (*i.e.,* flooding) query over our annotated overlay topology, initiating queries from 100 randomly selected peers for a particular target file and measuring the number messages needed to find the file. If significant topological clustering exists, then a few searches will complete using very few messages (*i.e.,* fewer hops), while most searches will require a much larger number of messages (*i.e.,* many more hops).

It is worth noticing that our annotated overlay topology is incomplete because of the existence of firewalled and departed peers. Hence, our calculated messages and hops are actually larger than they would be in practice. However, these absences do not significant affect topological clustering; if a file is common only in a certain neighborhood, that property is preserved even if information is absent for some fraction of the peers.

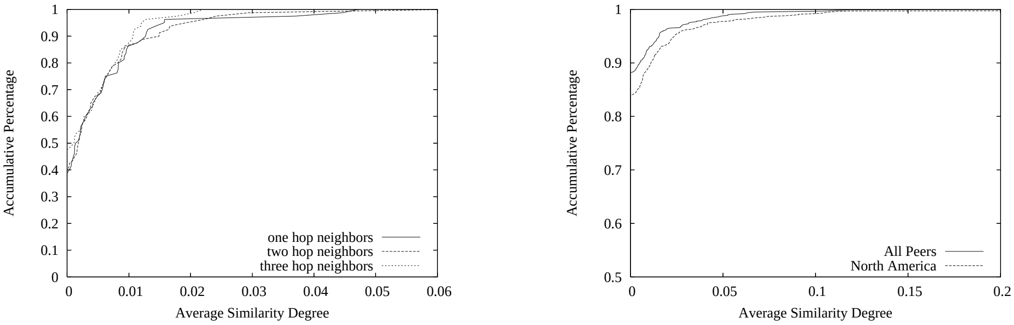
Based on our file popularity analysis, we selected five particular files as search targets, ranging from popular files (present on 5% of contacted peers) to unpopular files (present on 0.002% of contacted peers). For each target, the minimum number of flooding messages to locate at least 5 copies of the file is recorded and shown as CDF graph in Figure 5(a). Each line in the graph corresponds to a target file, with the file population ranging from 21,723 copies to 683 copies. More search messages are required for less popular files. The abrupt steps in the lines reflects the increase of one hop in each step of the expanding ring. For each target file, the CDF is very steep with nearly all queries completing in the same number of hops, or within a range of two hops (*e.g.,* the most popular file can be found in one hop 25% of the time and otherwise in two hops). This suggests files are evenly distributed.

To verify this finding, as a control we randomized the placement of files, guaranteeing that there was no topological clustering. The required flooding messages to reach 5 copies of the target files are plotted in Figure 5(b). The high similarity of the two figures indicates that significant topological clustering does not occur.

As a second test, we randomly chose a peer and calculated the average *similarity degree* between the peer and all its one-hop, two-hop, and three-hop neighbors. The similarity degree between two peers is defined as the number of common files divided by the smaller size of the sharing lists of the two peers. Figure 6(a) shows the CDF graph of the similarity degrees for 100 randomly chosen peers and their neighbors. If topological clustering were present, the similarity degree would decrease as the number of hops increased. However, as shown in the figure, there is not much difference in the similarity degree for neighbors at different distances.

The absence of topological clustering effect is most likely caused by the rapid churn of the Gnutella network topology. Peers join and leave the overlay frequently. In our prior work, we observed that given a snapshot, more than half of the Gnutella peers in the snapshot will have departed after 5 hours [11]. Each time when a peer joins the overlay, it attaches itself as a leaf peer at several random ultrapeers. Qualified leaf peers may become ultrapeers in order to maintain a proper ultrapeer-to-leaf ratio. Hence, the rapidly changing overlay topology prevents topological correlations from forming.

This finding is important for two reasons:



(a) Similarity degree distribution between one-hop, two-hop and three-hop neighbors

(b) Similarity degree between any two peers versus similarity degree between two peers from the same geographic area

Fig. 6. Cluster CDFs

- Measurement studies may sample the list of files from peers in any neighborhood in the overlay topology, without needing to capture the entire network
- Simulation studies may randomly distribute files among peers regardless of location. However, the number of files per peer should still follow a Zipf distribution.

While previous studies have frequently assumed these properties, to our knowledge they had not previously been empirically verified.

B. Geographic Clustering

Our second hypothesis is that the copies of a particular file tend to cluster based on the geographical location of peers. That is, the similarity degree of two peers in the same geographic area might be higher than two peers from different geographic areas, as a result of cultural preferences or language barriers. We use a coarse-grained approach to locate peers based on issuing a *whois* query to locate the Regional Internet Registry (RIR) responsible for their address. While not fine-grained, we expect this approach to be very accurate since errors should only be introduced when a peer is using some kind of IP tunneling.

To test for geographic clustering, we explore the degree of similarity between peers globally with the degree of similarity for one particular RIR: the American Registry for Internet Numbers (ARIN). We selected ARIN because it covers a smaller number of countries than other RIRs, which suggests any cultural clustering may be more apparent. ARIN includes Canada, the United States, and several countries in the Caribbean.

We calculated the similarity degree of 100 random Gnutella peer pairs drawn from ARIN and compared it with the similarity degree of 1000 peer pairs from all over the world. The CDF graph shown in Figure 6(b), shows a slightly higher similarity degree for peers in North America, demonstrating that file preferences are indeed correlated with geography.

VII. Dynamics Analysis

Section V examined static properties of files in a single snapshot. In this section, we explore dynamic properties of file popularity as it changes over time. Prior studies [5, 16] examine the change in the popularity of queries and file transfers in P2P file-sharing networks. However, to the best of our knowledge, no study has previously examined the change in the popularity of files actually stored in the network.

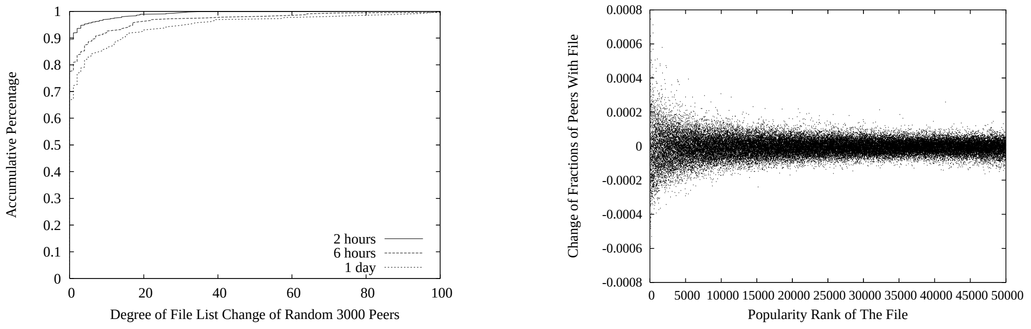
The key questions we are trying to answer are:

- How rapidly do peers change the set of files they are sharing?
- Do popular files experience greater, or smaller, shifts in popularity compared to less-popular files?
- Do changes in a particular file's popularity tend to follow a trend, or are they essentially random?

To capture the dynamic properties of shared file, we observed the Gnutella network for 9 consecutive days, from June 8th to June 16th, 2005. Additionally, we have a snapshot from May 5th, 2005. When we compare snapshots from different days, each snapshot was taken at approximately the same time of day.

A. Changes in files available at a particular peer

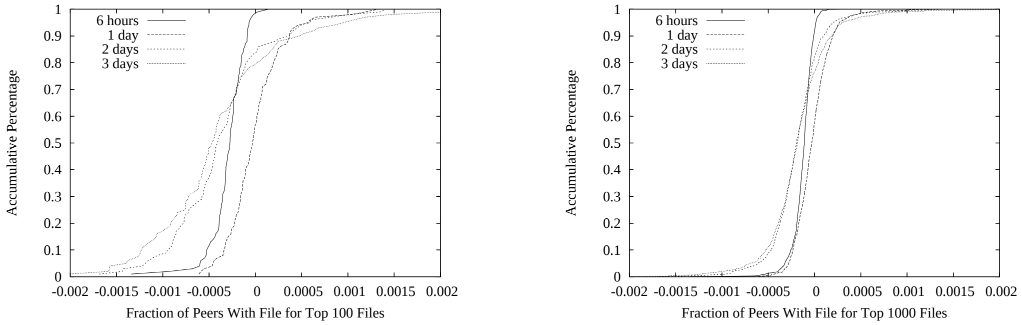
For a particular peer in the Gnutella network, there are two types of change that can occur to the list of files shared. First, the user may add new files, either by downloading them from other peers or by manually adding them to the shared folder. Second, the user may remove files, either by moving them out of the sharing folder or by deleting them entirely. Error may



(a) File List Change of Random 3000 Peers

(b) Popularity change versus file rank

Fig. 7. Changes in popularity



(a) Top 100

(b) Top 1000

Fig. 8. Popularity change of top N files

be introduced into our results if a peer that uses a dynamic IP address departs and another peer happens to take the same IP address. However, our prior work on churn [11] suggests this does not occur often. To create a single metric to unify both types of change (addition and deletion), we use the term *degree of change* to refer to the sum of the number of additions and deletions.

Figure 7(a) shows the CDF graph of the degree of change for a random selection of 3,000 peers, over intervals of 2 hours, 6 hours, and 1 day. As shown in the graph, during a 1 day period, 68% of peers keep the sharing list exactly the same and 90% of peers have a degree of change fewer than 20 files.

During a 2 hour period, 90% of peers do not change the sharing list, while 98% have a degree of change fewer than 20. Intuitively, for longer intervals, the degrees of change is generally larger. A small portion (about 1%) of peers change their sharing list by more than 100 files during the 6 hours and 1 day intervals.

These relatively small numbers, compared to the hundreds of files shared on average by each peer, suggests that caching information about the files shared by other peers can be a highly effective bandwidth-saving strategy in peer-to-peer systems.

B. *Changes in a file's popularity across all peers*

To eliminate the effects of a varying peer population, we define the *popularity* of a file as the fraction of successfully contacted peers with the file. Given the random distribution of files, the popularity can be interpreted as the probability of having that file for a random peer. We define the *change in popularity* as the difference between it's popularity at the beginning of an interval and it's popularity at the end of the interval.

In Figure 7(b), we plot the change of popularity for a particular file versus the rank of that file, during a 1 day interval. Each dot in the graph corresponds to a particular file. The population of the rank 50,000 file is about 100 in our measurement data. As shown in Figure 7(b), the top 15,000 files have significantly larger population changes than the rest of the files. However, the changes of popularity for most of the files are within 0.0002. For the unpopular files with rank more than 15,000, most of the changes are less than 0.0001. In summary, *popular files experience further variation in their popularity than unpopular files*.

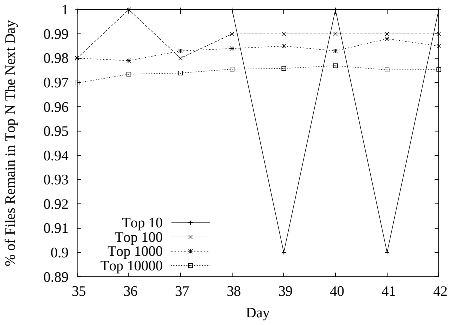
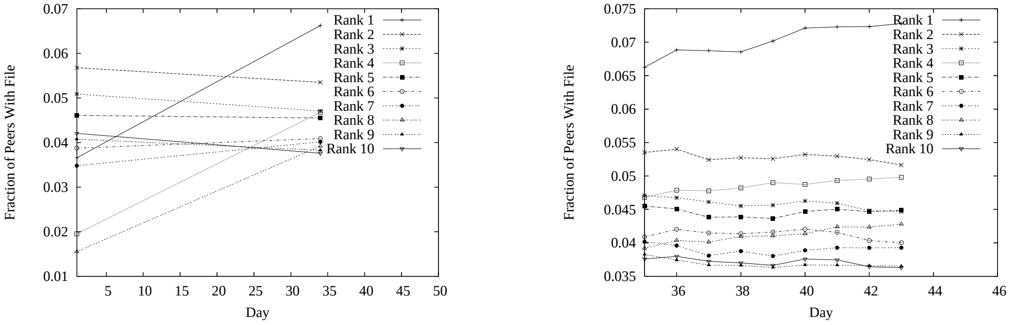


Fig. 9. Drift in Top-N popular file list



(a) 34 Days

(b) 9 Consecutive Days

Fig. 10. Consecutive popularity of top 10 files

To study this phenomenon in further detail, we zoom in on the top-100 and top-1000 most popular files and examine their change in popularity over different timescales. Figures 8(a) and 8(b) show the CDF graph of the change of popularity for the top-100 and top-1000 files, respectively, for intervals of 6 hours, 1 day, 2 days, and 3 days. Again, the changes are more dramatic over longer time scales. Additionally, comparing Figures 8(a) and 8(b) shows that more popular files exhibit greater variation in their popularity.

C. Trends in popularity variations

To study the trends in popularity change for particular files, we tracked the top-10 most popular files in our measurements. Figure 10(a) shows the shift in popularity from our May 5th snapshot to our June 8th snapshot[4]. While most files remained relatively steady, three files (ranks 1, 4, and 8) had a significant increase in their popularity. Figure 10(b) plots the popularity during the 9 following days, starting June 8th. The same three files continued their upward trend during this interval. These observations suggest that at least for popular files, shifts in popularity tend to follow trends rather than completely random changes.

Looking at this from a different angle, we examine how the list of the top-N files changed over the course of our measurements. Figure 9 shows the percentage of files of Day X that remain in the top-N list the next day. Note that the y-axis begins at 89%, indicating that the top-N list is highly stable from one day to the next. While the top-10 list appears to undergo more dramatic shifts, this is because a 1-file change means that only 90% of the files are still the same. For any of the top 10,000 popular files, the probability that the file remains in top 10,000 for the next day is more than 97%. Thus, we can conclude that the set of popular files in file-sharing networks exhibits little change over the course of a few days.

VIII. Conclusion

This paper presented a measurement-based characterization of available files in Gnutella file sharing application. We discussed the challenges in capturing an accurate snapshot of available files in P2P file-sharing applications, and then developed a

[4] We did initially capture data during the intermediate days. Unfortunately, we later discovered a bug which made these particular snapshots inaccurate. Therefore, we do not present that data. We plan to repeat this experiment and report more detailed results in the final paper.

new measurement methodology to achieve this goal. We use our parallel crawl to obtain fairly accurate snapshots of available files across peers in the Gnutella network along with the connectivity among peers. Using these snapshots, we conducted three types of analysis and provided a better understanding of distribution, correlation and dynamics of available files throughout the system.

We plan to continue this work in the following directions: We are currently collecting many more snapshots to repeat our analysis, gain more confidence to our findings, and investigate possible trends over longer timescales. Furthermore, we plan to develop and empirically evaluate various sampling techniques for monitoring different properties of available files without crawling the entire system.

References

[1] "slyck.com," http://www.slyck.com, 2005.

[2] S. Saroiu, K. P. Gummadi, R. J. Dunn, S. D. Gribble, and H. M. Levy, "An analysis of Internet content delivery systems," in *Symposium on Operating Systems Design and Implementation*, 2002, pp. 315–327.

[3] Daniel Stutzbach, Reza Rejaie, and Subhabrata Sen, "Characterizing Two-Tier Overlay Topologies in Modern P2P File-Sharing Systems," Tech. Rep. CIS-TR-2005-01, University of Oregon, Eugene, OR, Feb. 2005.

[4] Matei Ripeanu, Ian Foster, and Adriana 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.

[5] Alexander Klemm, Christoph Lindemann, Mary Vernon, and Oliver P. Waldhorst, "Characterizing the Query Behavior in Peer-to-Peer File Sharing Systems," in *Internet Measurement Conference*, Taormina, Italy, Oct. 2004.

[6] F. Le Fessant, S. Handurukande, A.-M. Kermarrec, and L. Massoulie, "Clustering in Peer-to-Peer File Sharing Workloads," in *International Workshop on Peer-to-Peer Systems*, 2004.

[7] Jian Liang, Rakesh Kumar, Yongjian Xi, and Keith W. Ross, "Pollution in P2P File Sharing Systems," in *INFOCOM*, Miami, FL, Mar. 2005.

[8] Jacky Chu, Kevin Labonte, and Brian Neil Levine, "Availability and Locality Measurements of Peer-to-Peer File Systems," in *ITCom: Scalability and Traffic Control in IP Networks II Conferences*, July 2002.

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

[10] Daniel Stutzbach and Reza Rejaie, "Capturing Accurate Snapshots of the Gnutella Network," in *Global Internet Symposium*, Miami, FL, Mar. 2005, pp. 127–132.

[11] Daniel Stutzbach and Reza Rejaie, "Characterizing Churn in Peer-to-Peer Networks," Tech. Rep. 2005-03, University of Oregon, Eugene, OR, May 2005.

[12] Ranjita Bhagwan, Stefan Savage, and Geoffrey Voelker, "Understanding Availability," in *International Workshop on Peer-to-Peer Systems*, 2003.

[13] Daniel Stutzbach and Reza Rejaie, "Characterizing the Two-Tier Gnutella Topology," in *SIGMETRICS*, Banff, AB, Canada, June 2005, Extended Abstract.

[14] Lada A. Adamic, Rajan M. Lukose, Bernardo Huberman, and Amit R. Puniyani, "Search in Power-Law Networks," *Physical Review E*, vol. 64, no. 46135, 2001.

[15] Krishna P. Gummadi, Richard J. Dunn, Stefan Saroiu, Steven D. Gribble, Henry M. Levy, and John Zahorjan, "Measurement, Modeling, and Analysis of a Peer-to-Peer File-Sharing Workload," in *SOSP*, 2003.

[16] Nathaniel Leibowitz, Matei Ripeanu, and Adam Wierzbicki, "Deconstructing the Kazaa Network," in *WIAPP*, 2003.

[17] Subhabrata Sen and Jia Wang, "Analyzing Peer-To-Peer Traffic Across Large Networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 2, pp. 219–232, Apr. 2004.

[18] Kunwadee Sripanidkulchai, Bruce Maggs, and Hui Zhang, "Efficient Content Location Using Interest-Based Locality in Peer-to-Peer Systems," in *INFOCOM*, 2003.

[19] Gnutella Developers Forum, "Browse Host Extension," http://www.the-gdf.org/wiki/index.php?title=Browse_Host_Extension.

[20] Daniel Stutzbach and Reza Rejaie, "Evaluating the Accuracy of Captured Snapshots by Peer-to-Peer Crawlers," in *Passive and Active Measurement Workshop*, Boston, MA, Mar. 2005, Extended Abstract, pp. 353–357.

[21] BearShare.com, "BearShare Network Statistics," http://www.bearshare.com/stats/, July 2005.

[22] E. Adar and B. A. Huberman, "Free riding on gnutella," *First Monday*, vol. 5, no. 10, Oct. 2000.