Characterizing Files in the Modern Gnutella Network: A Measurement Study

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

The Internet has witnessed an explosive increase in the popularity of Peer-to-Peer (P2P) file-sharing applications during the past few years. As these applications become more popular, it becomes increasingly important to characterize their behavior in order to improve their performance and quantify 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 the Gnutella system and conducted three types of analysis on available files: *(i)* Static analysis, *(ii)* Topological analysis and *(iii)* Dynamic analysis. Our results reveal several interesting properties of available files in Gnutella that can be leveraged to improve the design and evaluations of P2P file-sharing applications.

1. INTRODUCTION

During the past few years, the Internet has witnessed an explosive increase in the popularity of Peer-to-Peer (P2P) file sharing applications. 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 significantly contribute to network traffic [2]. These applications are primarily used for exchanging multimedia files where each participating 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 increasingly important because it reveals performance limitations of these applications in practice as well as 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 contributed resources (*i.e.,* storage space and available files) by individual users in 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 applications [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 applications (*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 along with new search mechanism (*e.g.,* dynamics querying in Gnutella) to improve their scalability. However, the potential effect of these new features on the characteristics of files has not been studied. Third, previous studies have not examined the dynamics of file characteristics over time nor any possible correlation between the overlay topology and file distribution.

In this paper, we empirically characterize available files across all the reachable peers in the modern Gnutella network. We present a new measurement methodology to capture an accurate "snapshot" of the system at a particular point of time. A snapshot contains participating peers in the system, available files at each peer and the pair-wise connectivity among peers (*i.e.,* the overlay topology). We have developed a fast P2P crawler called Cruiser [10]. Using Cruiser, we have captured

around 40 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 on shared files in Gnutella: **1) Static Analysis:** We examine properties of contributed resources (*i.e.,* files, and storage space) by participating peers in individual snapshots of the system. **2) Topological Analysis:** We investigate any potential correlation between the pattern of file distribution among peers and the overlay topology. **3) Dynamic Analysis:** We study variations in the popularity of individual files among peers over different timescales.

Our main findings can be summarized as follows: *(i)* 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. *(ii)* The number of shared files and contributed storage space by individual peers both follow a power-law distribution. Compared to earlier studies, each Gnutella user contributes significantly more disk space but shares approximately the same number of files. *(iii)* The popularity of individual files follows a Zipf distribution. This implies that a small number of files are extremely popular but a majority of files are very unpopular. *(iv)* The most popular file type is the MP3 file, which accounts 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. Furthermore, 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. *(v)* Files are randomly distributed throughout the overlay and there is no strong correlation between the available files at peers that are one, two or three hops apart in the overlay topology. *(vi)* 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.

Why Characterize Gnutella? We conducted 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, 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 easily extracted from a peer. This eliminates the need for reverse-engineering the protocol which 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 3 describes the challenges in capturing accurate snapshots and describes our measurement methodology and tools. Section 4, 5 and 6 present static analysis, topological analysis, and dynamics analysis of files in the Gnutella network, respectively. Section 7 provides an overview of previous studies. Finally, Section 8 concludes the paper and sketches our future plans.

2. 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 (especially 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 system. In the original Gnutella protocol, participating peers form an unstructured overlay topology that is used by each peer to perform flood-based searches among other peers. To improve the scalability of the original 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. Second, we use Gnutella's Browse-Host extension [11] to acquire the list of files shared by each peer. In the following two subsections, we further elaborate on these two issues.

Two-Tier Topology: As shown in Figure 1, modern Gnutella clients implement a two-tiered overlay structure by dividing peers into two groups: *ultrapeers* (or super-peers) and *leaf peers*. Each ultrapeer neighbors with several other ultrapeers to form 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 for its filename keywords to that ultrapeer. This allows the ultrapeer to only forward messages to the leaves that

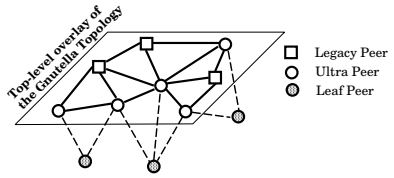


Figure 1. Two-Tier Topology of Modern Gnutella

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.

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 [12]. The Browse-Host extension [11] is an extension of Gnutella protocol that enables one peer to view the list of files shared (called a *sharing list*) by another peer. This feature allows 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 to use Browse-Host for capturing the files shared by most peers in Gnutella.

3. MEASUREMENT METHODOLOGY

Our goal is to capture a *snapshot* of the Gnutella network at a given point of time which contains *(i)* all participating peers and the pair-wise connectivity among them (*i.e.,* an overlay snapshot), and *(ii)* available files at each participating peer in the overlay (*i.e.,* a file snapshot). In essence, we need to capture snapshots of the overlay topology and annotate each peer with its available files. A common approach to capture a snapshot is to use a P2P crawler. Given a set of initial peers, a crawler contacts individual peers to capture their available files and collect information about their neighboring 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 large size and dynamic nature of participating peers (or churn) coupled with the slow speed of crawlers in practice, captured snapshots by a crawler are inherently *distorted* [13]. More specifically, as the crawler explores the overlay, many peers join or leave the system which results in changes in the overlay topology and possibly changes in the set of available files throughout the system. Therefore, the captured snapshot contains a group of peers that have departed or arrived 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*.

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] periodically capture a small subset of participating peers (*i.e.,* a partial snapshot) through a short crawl and assume that the captured peers properly represent the entire population. *(ii) Periodic Probe of a Fixed Group*: Other studies identify a subset of some participating peers (using a partial crawl or passive monitoring) and periodically probe the same group of peers to collect information about their available files [8]. In the absence of any solid understanding of file characteristics in P2P systems 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 the Gnutella network. Our goal is to capture the entire population of participating peers in the Gnutella network (*i.e.,* a complete snapshot) within a short period to minimize any potential bias in our characterization. Note that the time required to obtain the list of available files at individual peers is significantly longer than for obtaining 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. In a nutshell, a topology crawl is significantly faster than a content crawl. Therefore, we decouple topology and content crawls to improve the accuracy of captured snapshots and conduct our snapshots in three steps as follows: First, we conduct a *topology crawl* to quickly capture all participating peers and their pair-wise connectivity, *i.e.,*

*In the extreme case, the crawler may never terminate since there are always new peers to contact.

Crawl Date	Type	Number	TCP Refused	Timeout	Conn. Lost	App. Refused
Jun. 13	Ultrapeers	281,472	34.90%	3.48%	2.97%	2.18%
	Leaf Peers	1,932,944	85.39%	0.90%	0.88%	0.55%
	Total Peers	2,218,883	78.93%	1.23%	1.15%	0.76%
Aug. 31	Ultrapeers	347,168	35.70%	3.76%	6.35%	2.03%
	Leaf Peers	2,320,088	86.98%	0.86%	1.08%	0.46%
	Total	2,674,452	80.27%	1.24%	1.82%	0.67%
Oct. 13	Ultrapeers	320,063	36.05%	3.47%	5.89%	2.25%
	Leaf Peers	2,790,171	87.27%	0.84%	1.04%	0.37%
	Total	3,118,986	81.96%	1.11%	1.59%	0.56%

Table 1. Statistics on Sample Snapshots

capturing the overlay topology. Second, we conduct a *content crawl* and collect the list of files available at each one of the peers identified during the topology crawl. Third, once the content crawl is completed, we conduct another topology crawl in order to identify those long-lived peers in the initial snapshots that remained in the system during the entire content crawl. 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. Since some of the captured peers in the first topology crawl depart the system during the content crawl, the collected content in our measurement is slightly biased towards peers with longer uptime.

Cruiser, A Parallel Crawler: To capture accurate snapshots of P2P systems, we have developed a parallel P2P crawler, called Cruiser [10], that can crawl an overlay orders of magnitude faster than any previous crawler. Cruiser achieves this goal by significantly increasing the degree of concurrency in 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, (15min + 5.5hr + 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: We have captured around 40 snapshots of the Gnutella network annotated with the list of files available at each peer in three measurement periods. Each period consists of daily snapshots from consecutive days during the following intervals: 6/8/2005–6/18/2005, 8/23/2005–9/9/2005 and 10/11/2005–10/21/2005[†]. These snapshots enable us to examine characteristics of available files in the system over both short and long timescales (*i.e.,* several days and several months). Table 1 summarizes the statistics for several sample snapshots. As shown in this table, we divide captured peers into two groups: Ultrapeers and Leaf peers. Legacy peers constitute a negligible portion (less than 0.3%) of captured peers in each snapshots and thus they are omitted. A subset of peers in each group might be unreachable by our crawlers for one of the following reasons: *(i)* firewall or NAT blocking incoming traffic, *(ii)* severe network congestion or overloaded processor at the peer, *(iii)* the peer departed, or *(iv)* or the peer does not support the Browse Host protocol. Since ultrapeers are not allowed to be firewalled, any reported connection error for ultrapeers indicates that the contacted peer has departed. However, connection errors for leaf peers might occur due to peer departure or a firewall[‡]. In our earlier study of Gnutella [3], we showed that about half of all leaf peers leave the overlay within a 5 hour period. Independent online statistics [14] 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 the column labeled as "TCP Refused"). 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 primarily due to two reasons: *(i)* the ratio of departed peers during the long content crawl, and *(ii)* the large number of leaf peers behind firewalls.

We briefly discuss several interesting problems that we experienced during our data collection and data processing.
Low-bandwidth TCP Connection: Although Cruiser has a timeout mechanism that closes any idle connections after 20

[†]We are missing snapshots for 9/6/2005, 9/8/2005, 10/17/2005, and 10/29/2005 due to a mixture of software, network, power failures.
[‡]We are not aware of any reliable technique to distinguish between these two scenarios.

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 rate (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 individual files. In our initial measurements, we observed many files with the same name 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 (*e.g.,* [8]), is not a reliable file 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 where each segment contains a random portion of the peers. Then, we calculated the popularity of files within each segment, trimmed all files that had less than 10 copies in a segment, and combined all the trimmed segments into a single snapshot. This approximation eliminated several million distinct files and prevented memory bottlenecks during our post-processing. While this prevented us from performing analysis on the least popular files (with less than 70 copies in the entire network), it should not affect conducted analysis on more popular files.

4. STATIC ANALYSIS

In this section, we examine characteristics of available files across all peers in individual snapshots regardless of their location in the overlay topology. In particular, we examine the following characteristics: *(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 these characteristics of the Gnutella network over the past few years. To allow cross-comparison of different results in this section, we mostly focus the three candidate snapshots listed in Table 1. However, we have examined several other snapshots and observed similar behavior. Therefore, the presented results provide representative behavior across our snapshots.

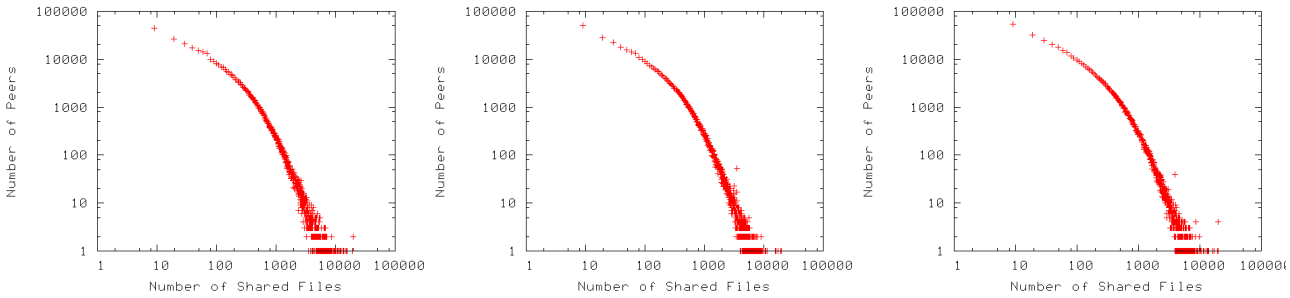
Ratio of Free Riders: The success of P2P file sharing systems depends on the willingness of participating peers to share their files. However, previous studies have frequently reported that participating peers do not have an incentive to contribute their resources (*e.g.,* disk space and network bandwidth) to the system and thus only use resources offered by others, *i.e.,* become "free riders". In particular, Adar *et al.* [15] 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 2 presents the degree of free riding among Gnutella peers in the three candidate snapshots. 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 3. For each one of the above groups, the corresponding row in Table 2 presents *(i)* the number of cooperative peers that provided their sharing list (labeled as "Peers"), *(ii)* the ratio of free riders (labeled as "None"), and *(iii)* the average number of shared files by each peer for each candidate snapshot (labeled as "Files"). Table 2 shows several interesting points as follows: First, the ratio of free riders in Gnutella has significantly dropped from 25% in 2002 to around 13% among all participating peers (*i.e.,* last row), and it is drastically lower than the 68% recently reported in eDonkey [6]. We speculate that several factors have contributed in the observed drop in the ratio of free riders including the increase in access link bandwidth for average Internet users and marketing efforts by the Gnutella vendors encouraging their users to share. Second, Table 2 reveals that the ratio of free riding among ultrapeers (10-12%) is somewhat lower than that in leaf peers (14.5-16%). However, since leaf peers constitute a larger portion of the total population (86–90%), their behavior has a bigger impact on system performance. Furthermore long-lived peers have a slightly higher ratio of free riders (especially among leaf peers) compared to short-lived peers in the same group. Third, the average number of shared files indicates that the user's sharing behavior does not strongly correlate with their uptime or peer type. Other captured snapshots exhibited similar trends in the ratio of free riding and mean number of shared files.

Degree of Resource Sharing Among Cooperative Peers: We now turn our attention to cooperative peers and characterize their willingness to contribute their resources (*i.e.,* both files and storage space). During our analysis, we noticed that the

	June 13th, 2005			August 31st, 2005			October 13th, 2005		
	Peers	**None**	**Files**	**Peers**	**None**	**Files**	**Peers**	**None**	**Files**
Ultra	158,993	12.11%	352	181,052	10.08%	347	167,512	10.19%	344
Leaf	234,812	15.49%	332	242,524	15.85%	380	286,478	14.49%	371
Long-lived Ultra	124,762	12.27%	349	141,823	10.26%	343	132,219	10.40%	341
Short-lived Ultra	34,231	11.52%	363	39,229	9.40%	358	35,293	9.40%	356
Long-lived Leaf	155,999	16.26%	350	158,965	16.93%	405	171,384	15.99%	406
Short-lived Leaf	78,813	13.94%	297	83,559	13.79%	335	115,094	12.24%	320
Total	393,805	14.15%	340	423,576	13.41%	365	453,990	12.91%	360

Table 2. Fraction of peers free-riding and mean shared files per peer



(a) June 13th, 2005 (b) August 31st, 2005 (c) October 13th, 2005

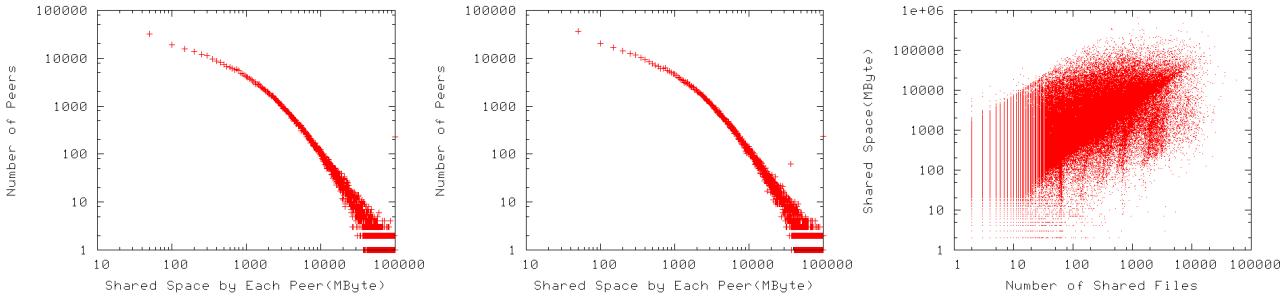
Figure 2. Distribution of the number of shared files

sharing lists of many peers contain duplicate files. This occurs because most Gnutella clients simply put various folders with potentially duplicate content under the sharing folder. We have excluded all the duplicate files from captured sharing lists (which account for around 10 million files or roughly 9% of all captured files) in all the reported analysis in this paper. Figure 2 plots the distribution of the number of peers that are willing to share x files in three candidate snapshots. This distribution is very similar across all three snapshots and largely conforms to a power-law, though somewhat lower for peers sharing few files. To illustrate a different angle of resource sharing, Figures 3(a) and 3(b) depict the distribution of contributed disk space (in megabytes) among cooperative peers in two candidate snapshots. This distribution is also very similar across both snapshots and appears to follow a power law distribution. This indicates that most of the participating peers contribute little disk space ($<$ 100 MB) while a small number of peers contribute a very large storage space (50–100 GB).

Saroiu *et al.* [9] reported a strong correlation between the number of shared files and the volume of shared data across Gnutella peers in 2002. Figure 3(c) shows this correlation as a scatter-plot across all cooperative peers in one snapshot (June 13th, 2005). Each point in this figure represents the number of shared files versus the shared disk space for each cooperative 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, likely due to the rise in the popularity of very large video files. There is a discernable line with the slope around 3.7MB/file in Figure 3(c) which is the typical size of a MP3 audio file.

File Popularity Distribution: The distribution of popularity for individual files throughout the system is an important property that shows the degree of similarity and thus availability of individual files among participating peers. 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. Furthermore, Fessant *et al.* [6] recently reported a Zipf distribution for the file popularity in eDonkey. However, none of these studies have captured a large number of peers.

Figure 4 shows the distribution of file popularity as a function of its rank (in log-log scale) from a random subset of cooperative peers (as we described in Section 3). in Gnutella across three candidate snapshots. In total, each snapshot contains more than 800 terabytes worth of content in more than 100 million unique files based on information from 0.4 million peers, constituting 18.5% of identified peers. If we assume that unreachable peers have similar profiles, the volume

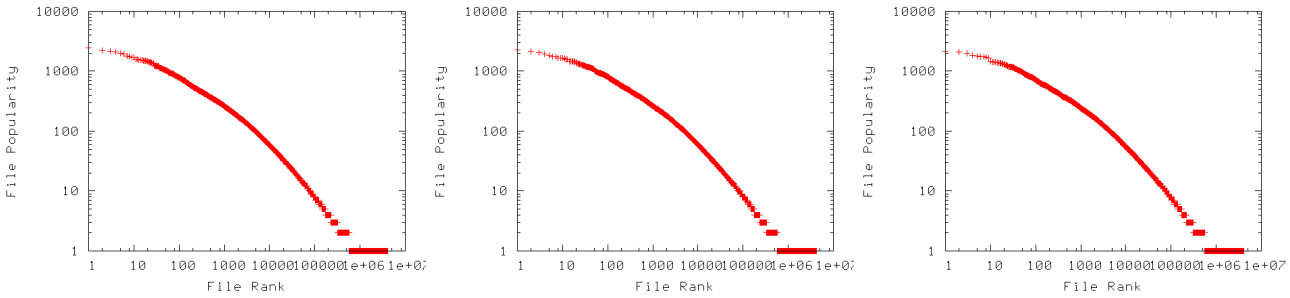


(a) Distribution of shared bytes, June 13th, 2005

(b) Distribution of shared bytes, October 13th, 2005

(c) Correlation between shared bytes and number of shared files, June 13th, 2005

Figure 3. Distribution of the number of shared files



(a) June 13th, 2005

(b) August 31st, 2005

(c) October 13th, 2005

Figure 4. Distribution of file popularity from a random subset of peers in each snapshot

of available content in the Gnutella network is around 4,400 terabytes. Figure 4 illustrates two points: *(i)* file popularity mostly follows a Zipf distribution, and *(ii)* the distribution of file popularity has remained very stable across the four month measurement period. The Zipf distribution of file popularity implies that a small number of files are extremely popular while only a few copies are available for many other files among participating peers. Examination of other snapshots revealed that file popularity exhibits the same distribution across different snapshots.

File Type Analysis: We have also examined the distribution of available files among Gnutella peers across different types of video and audio formats. This basically illustrates what types of content are available in the system and thus exchanged among peers. 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. However, video files were significantly less popular and only contributed 2.1% of files and 19.1% of bytes.

Using our snapshots, we analyze the various types of audio and video files based on file extensions. Table 3(a) and 3(b) list the top ten most popular file types (in terms of number of files), along with their popularity and their contribution in the available content across cooperative Gnutella peers in two snapshots that are four months apart. The distribution of file types in these two snapshots seems fairly consistent with respect to both files and bytes. Note that except for mp3 and jpg files, the percentage of other file types are rather small and very close. Therefore, their ranking could easily change across different snapshots due to the variations in the identity of participating peers. Table 3(a) and 3(b) show that mp3 audio files are significantly more popular than any other file type and occupy more than one third of all disk space in the system.

Table 3(c) shows the range of popularity for the most popular audio and video file types across nine consecutive snapshots in June of 2005. Although non-media files (*i.e.,* jpg, gif, htm, exe, txt) are among the top ten most popular types, audio and video files (*e.g.,* mp3, mpg, avi) collectively occupy more than 93% of bytes and make up more than 73% of all files in the system. This table also reveals that mpg files are significantly larger than other file types in the system. 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 (*i.e.,* most files are audio files, but most bytes are in video files). Comparing to the reported results by Chu at al. in 2001 [8], video files have become three times

Type	Files	Bytes
mp3	61.5%	37.80%
jpg	7.54%	0.213%
gif	3.14%	0.003%
wma	2.70%	1.283%
htm	2.69%	0.004%
exe	2.65%	0.597%
wmv	2.62%	3.413%
mpg	1.91%	21.51%
wav	1.86%	0.706%
txt	1.62%	0.005%
Total	88.2%	65.53%

(a) Top 10 Most Popular File Types, June 13th, 2005

Type	Files	Bytes
mp3	61.6%	36.41%
jpg	7.08%	0.231%
wma	2.65%	1.213%
asf	2.63%	0.443%
gif	2.63%	0.003%
exe	2.51%	0.468%
htm	2.42%	0.004%
mpg	2.03%	23.19%
m4a	1.67%	0.820%
wav	1.55%	0.583%
Total	86.8%	63.36%

(b) Top 10 Most Popular File Types, October 13th, 2005

Major Audio Types		
Type	**Files (%)**	**Bytes (%)**
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
Total	67.14-67.58	39.68-41.21
Major Video Types		
Type	**Files (%)**	**Bytes (%)**
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
Total	5.65-6.16	52.02-53.16

(c) Major audio and video file types during June 8th - June 16th

Table 3. Distribution of major audio and video file types

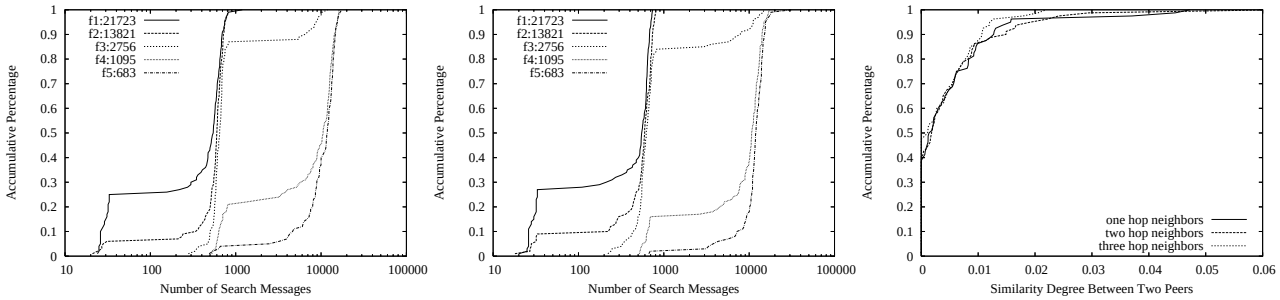
more popular and occupy almost three times more space in the system. This table also reveals that avi files contribute a significant portion of bytes (around 25%) while they constitute less than 1% of available files.

5. TOPOLOGICAL ANALYSIS

In this section, we investigate whether copies of a given file are located at close-by peers (*i.e., topologically clustered*) or at randomly scattered peers throughout the overlay. Understanding this issue would be useful in both the design and evaluations of search techniques in file-sharing applications. There are two factors that affect the location of available copies of individual files throughout the overlay: *(i)* scoped search, and *(ii)* churn. To accommodate scalability, each searching peer often examines available content among nearby To accommodate scalability, each searching peer often examines available files among nearby peers (*i.e.,* conducts scoped search). This suggests that a single copy of a file gradually diffuses outward from the original location to other nearby peers and some type of clustering might exist. However, due to the dynamics of peer participation (or churn), the location of a peer in the overlay changes over time and could prevent such clustering. The key question is which one of these two factors has the dominating effect. To answer this question, we generate an annotated snapshot of the overlay topology where each peer is annotated with its available files. Then, we explore topological clustering from the following two perspectives:

Per-File Perspective: We conduct trace-driven simulation of flood-based querying over our annotated overlay topology. Figure 5(a) depicts the CDF of the minimum number of query messages to find five copies of a target file from 100 randomly selected peers in the overlay. Each line in this figure corresponds to a target file with different popularity from our static analysis. The abrupt steps in the lines indicates one-hop increases in the scope of the search. Note that each CDF has at most two steps. This implies that most of the 100 randomly selected peers find all 5 copies among other peers that are either n or $n + 1$ hops away where n inversely depends on the log of the target file's popularity. Clearly, more search messages are required for less popular files. If significant topological clustering exists, 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). However, the pattern of increase in the number of search messages for less popular files in Figure 5(a) indicates that files are randomly distributed.

To verify this conclusion, we randomized the placement of available files, guaranteeing that no topological clustering exists in the overlay. Figure 5(b) shows the CDF of the number of required query messages to reach 5 copies of the same target files over the annotated topology with randomized file placement. The high similarity between Figures 5(b) and



(a) True file locations; Number of query messages to reach 5 copies of a file

(b) Randomized file locations; Number of query messages to reach 5 copies of a file

(c) Similarity degree distribution for peers one-hop, two-hops, and three-hops away

Figure 5. Distribution of file popularity from a random subset of peers in each snapshot

5(a) supports our conclusion that no significant topological clustering exists in the locations of a given file throughout the overlay. It is worth noting that our annotated overlay topology is incomplete due to the unreachable peers behind firewalls and departed peers. This implies that the number of messages are likely to be smaller in practice. However, the unreachable peers should not significantly affect topological clustering. More specifically, if a file is available only in a certain region of the overlay, the clustering property is preserved even if information is absent for some fraction of the peers in that region.

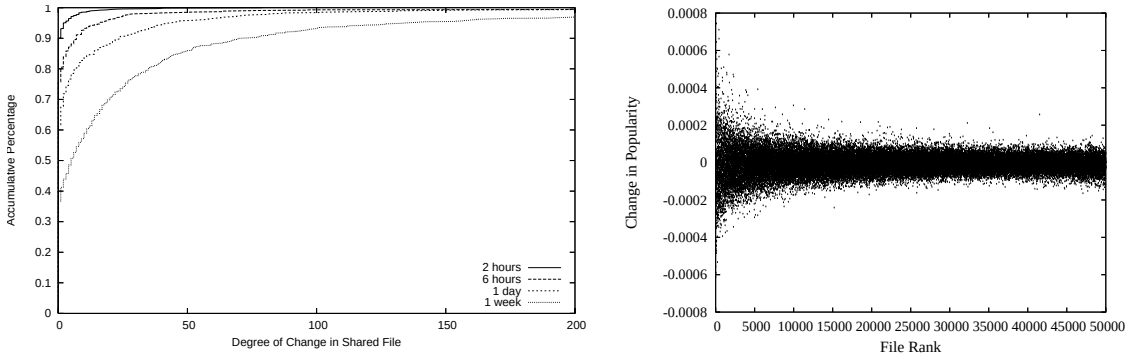
Per-Peer Perspective: We examine the similarity degree between available content at a random peer and all its one-hop, two-hop, and three-hop neighbors The similarity degree between two peers is defined as the number of common files divided by the smaller size of the sharing lists for the two peers. Figure 5(c) shows a separate CDF for the average similarity degrees of 100 randomly selected peers with their one-hop, two-hop and three-hop neighbors. If topological clustering were present, the similarity degree would decrease as the distance between comparing peers increases. However, since the distributions are nearly identical, Figure 5(c) clearly illustrates that there is no correlation between the similarity degree and distance between two peers.

These results suggest that churn is the dominant factor in determining the distribution of files throughout the overlay. In our prior work, we observed that more than half of the Gnutella peers in a given snapshot will depart within 5 hours [16]. Any time a peer joins the overlay, it attaches itself as a leaf peer at several random ultrapeers. Furthermore, qualified leaf peers may become ultrapeers in order to maintain a proper ultrapeer-to-leaf ratio. Hence, the rapid changes in the overlay topology prevents formation of such topological clustering. This finding is important for two reasons: *(i)* Measurement studies may sample the list of files from random peers in any part of the overlay topology, and do not need to capture the entire network. *(ii)* Simulation studies may randomly distribute available files among participating peers regardless of location. However, the number of files per peer should follow a Zipf distribution while the number of copies for each file should follow a power law as we showed in Section 4. While previous studies have frequently assumed these properties, to our knowledge, they had not previously been empirically verified.

6. DYNAMIC ANALYSIS

In this section, we turn our attention to the dynamic properties of available files in Gnutella. More specifically, we investigate how various properties of available files change over time. Prior studies examined changes in the popularity of queries and exchanged files in P2P file-sharing systems (*e.g.,* [5, 17]). However, to our knowledge, no study has previously explored the dynamic characteristics of stored files in P2P systems. For these analysis, we leverage our three sets of daily snapshots that were captured once every two months. This dataset allows us to explore dynamic properties over short timescales (*i.e.,* hours and days) as well as long timescales (*i.e.,* months). We explore the following three aspects of the dynamic properties of shared files: *(i)* variations in shared files by individual peers, *(ii)* variations in popularity of individual files, and *(iii)* predictability of future popularity.

Variations in Shared Files by Individual Peers: Our goal is to determine how rapidly the available files at individual peers change with time. These dynamics show whether past information about the available files at individual peers can be



(a) File List Change of Random 3000 Peers

(b) Change in popularity of files over a one day interval

Figure 6. Changes in popularity

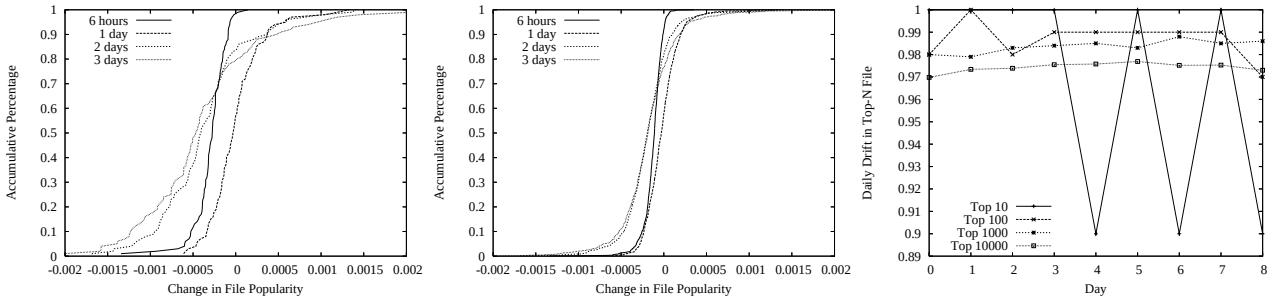
reliably used in future searches. There are two types of change that can occur to the list of shared files at each peer. First, the user may add new files, either by downloading them from other peers or by manually adding them to the shared folder. Second, the user may remove files, by moving (or deleting) those files from the sharing folder. We note that dynamic IP addresses may introduce error into our results when a peer departs from the system and another peer joins the system with the same IP address. However, our prior study on churn revealed that such events are not common [16]. We define the total number of added and deleted files at a single peer as the *degree of change* to quantify both types of changes in shared files at each peer.

Figure 6(a) depicts the CDF of the degree of change for 3000 randomly selected peers over four different timescales: 2 hours, 6 hours, 1 day and one week. This figure reveals that 36% of monitored peers do not change their sharing lists over a one week interval. However, that number increases to 69%, 80% and 90% over one day, 6 hour and 2 hour intervals, respectively. On the other hand, 90% of peers change less than 10 files over a 6 hour period, less than 25 files over a one day period, and less than 80 files over a week. Given the average number of shared files by each peer (which is around 350 from Table 2), this result indicates that the variations of shared files by individual peers is rather small especially over several days. While this finding is rather intuitive and expected, Figure 6(a) allows us to quantify the distribution of the degree of change across different timescales. Finally, this result implies that caching information about the available files at other peers (especially over the timescale of few days) can be a highly effective bandwidth-saving strategy in peer-to-peer systems.

Variations in Popularity of Individual Files: We focus on the effect of changes in shared files at each peer on the popularity of individual files across the system. Understanding the dynamics of file popularity can be used to determine how often the popularity of available files should be sampled. To eliminate the effects of a varying peer population across different snapshots, we define the *popularity* of a file as the fraction of successfully contacted peers with the file. Given the random distribution of files among peers, the popularity can be interpreted as the probability of having that file at a random peer. We define the *change in popularity* of a given file over interval τ as the difference between it's popularity at the beginning and the end of such an interval.

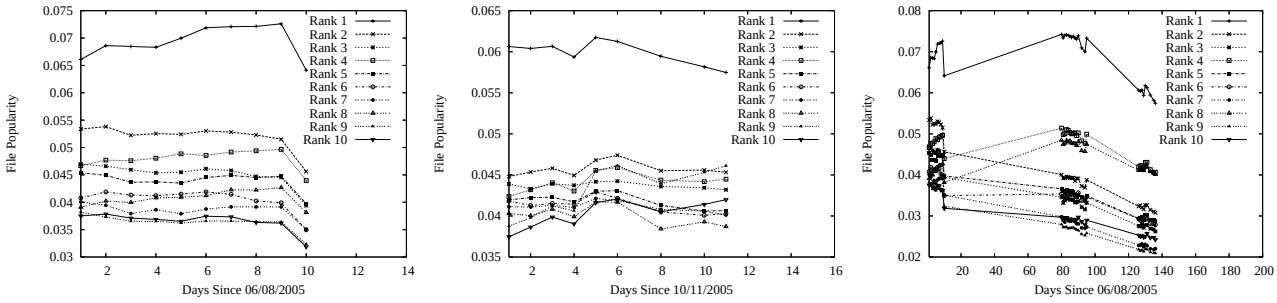
Figure 6(b) depicts the change in popularity of 50,000 files over a one day interval as a function of their popularity rank as a scattered-dot plot where each dot corresponds to a particular file. The population of the highest and lowest ranked files in this figure are roughly 28,000 and 130, respectively. This figure clearly demonstrates the effect of file popularity on the variations of its popularity over a one-day period. This figure shows that the most popular files (top 15,000) exhibit significantly larger variations (increase or decrease) in their popularity compared to the rest of the files. Note that the changes in popularity for most of the files are within 0.0002. While the variations in file popularity rapidly drop with file rank among the top 15,000 files, it becomes relatively stable across the least popular files and remains below 0.0001. In summary, a group of the most popular files experience wider variations in their popularity than unpopular files.

To study popularity dynamics in further detail, we focus on the top-100 and top-1000 most popular files and examine the popularity variations over different timescales. Figures 7(a) and 7(b) plot the CDF graph of the change of popularity



(a) Popularity change over top 100 files (b) Popularity change over top 1000 files (c) Drift in the Top-N popular file list

Figure 7. Changes in file popularity



(a) June 8th–June 17th, 2005 (b) October 11th–October 20th, 2005 (c) June 8th–October 20th, 2005

Figure 8. Changes in the popularity of the Top-10 files on different timescales

for the top-100 and top-1000 files, respectively, for intervals of 6 hours, 1 day, 2 days, and 3 days. Figures 7(a) and 7(b) individually show that the range of variations in popularity of top n files expanded with time. However, comparing these figures clearly illustrates that *(i)* for any given timescale, more popular files exhibit significantly larger variations in their popularity, and *(ii)* their popularity changes more rapidly with time.

Trends in Popularity Variations: The final question that we try to answer is whether variations in file popularity over time follow a certain trend. More specifically, can one predict the popularity of a given file in the near future based on the observed trend in popularity during the recent past? If such a correlation in popularity variations exists, then it can be leveraged to estimate the popularity of desired files and incorporate that information into sharing or search mechanisms.

Intuitively, the popularity of a new file should increase with some average rate until it reaches its peak popularity, and then gradually decrease. However, the rate and pattern of change in popularity, the range of maximum popularity and the time that a file remains at its peak could significantly vary across different files. To study the trends in the popularity changes of individual files, we tracked the popularity of the top-10 files across several days. Figures 8(a) and 8(b) show the variations in popularity of the top-10 files on 06/08/2005 and 10/11/2005 over the several proceeding days[§]. While the pattern of changes are different across these two figures, they both show that the popularity of the top-10 files remains relative stable across a few days. The noise-like variations in file popularity can be attributed to changes in the identity of participating peers during our measurement on different days.

We also investigated long-term trends in popularity changes over several months. Figure 8(c) depicts the popularity of the top-10 files on 6/8/2005 across our three set of consecutive snapshots, namely 6/8/2005-6/18/2005, 8/23/2005-9/9/2005 and 10/11/2005-10/21/2005. Note that the selected files are unlikely to remain in the top-10 list across all these snapshots. Figure 8(c) illustrates that observed changes in popularity of the top-10 files are more pronounced over an interval of a few months. Furthermore, it shows that the popularity of some of these files (namely ranks 1, 4 and 8) are initially increasing

[§]Note that the popularity information for a couple of days are missing as we indicated in Section 3.

and then decreasing whereas the popularity of other files is consistently decreasing over several months. In summary, while our results suggest that the popularity of the top-10 files remains relatively stable over several days, they exhibit more visible changes over several months. Unfortunately, we do not have sufficient data to identify more specific trends in popularity with sufficient confidence. This remains as a future work item for us to explore.

Looking at popularity from a different angle, we examine how the identity of the top-N files changes on a daily basis. Figure 7(c) shows the percentage of the top-N files on day $x-1$ (starting at 06/08/2005) that remain in the top-N files on day x (*i.e.,* daily drift in the top-N list) for four values of N. Note that the y-axis begins at 89%, indicating that the top-N list is highly stable from one day to the next. The top-10 list appears to undergo more dramatic shifts (*i.e.,* 10%). This is mainly due to two reasons: *(i)* the small number of files in the top-10 (one change in the list results in 10% variations) coupled with *(ii)* the noisy variations in popularity of individual files that can easily change a couple of files at the bottom of the top-10 list. The main conclusion from this figure is that the identity of the top-N list remains highly stable across consecutive days for different values of N.

7. RELATED WORK

Several measurement studies have examined different properties of P2P file-sharing networks including: *(i)* dynamics of peer participations (*i.e.,* churn) [16, 18], *(ii)* overlay topology structure [3, 4, 19, 20], *(iii)* query traffic [5], *(iv)* data traffic [17, 21, 22], and *(v)* shared files [6, 7]. We are aware of only two other studies that focus on the characteristics of shared files by users. First, Fessant *et al.* [6] examined characteristics of available 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. 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 showed 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. In 2001, Chu *et al.* [8] studied peer churn and the distribution of file popularity. They found that file popularity follows a log-quadratic distribution (which can be thought of as a second-order Zipf distribution). Saroiu *et al.* [9] examined many characteristics of peers in Napster and Gnutella, such as their bottleneck bandwidth, latency, uptime, and number of shared files in 2001. They found that the number of shared files was heavily skewed.

Our study differs from the few previous studies on the shared files in P2P systems at least in two ways. First, we used recent and accurate snapshots of the Gnutella network with a significantly larger population of peers (*i.e.,* more than a million concurrent peers). Second, we presented two properties of the shared files that have not previously been studied: *(i)* the correlation between shared files among peers and the overlay topology structure, and *(ii)* the variations in popularity of shared files across participating peers over time.

Another group of studies passively captured P2P traffic at a router to characterize exchanged files among peers. Gummadi *et al.* [21] analyzed a 200-day trace of Kazaa traffic collected at the University of Washington, demonstrating that file transfers in Kazaa do not follow a Zipf distribution and argued 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 *et al.* [17] at a large Israeli ISP. They examined the changing popularity of exchanged files among peers and showed that the data-sharing graph exhibits small-world properties. Note that the pattern of exchanged files among peers affects the characteristics of shared files throughout the system, but is subject to shorter-term trends. In contrast, the shared files by a peer may be the result of transfers over the course of months or years, followed by a gradual pruning of unwanted files. In summary, these studies on exchanged files are closely related and complement our work.

8. CONCLUSION

This paper presented a measurement-based characterization of available files in the 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 new measurement methodology to achieve this goal. We used 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 the 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 in 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*, pp. 315–327, 2002.
3. D. Stutzbach, R. Rejaie, and S. Sen, "Characterizing Unstructured Overlay Topologies in Modern P2P File-Sharing Systems," in *Internet Measurement Conference*, pp. 49–62, (Berkeley, CA), Oct. 2005.
4. M. Ripeanu, I. Foster, and A. Iamnitchi, "Mapping the Gnutella Network: Properties of Large-Scale Peer-to-Peer Systems and Implications for System Design," *IEEE Internet Computing Journal* **6**(1), 2002.
5. A. Klemm, C. Lindemann, M. Vernon, and O. P. Waldhorst, "Characterizing the Query Behavior in Peer-to-Peer File Sharing Systems," in *Internet Measurement Conference*, (Taormina, Italy), Oct. 2004.
6. F. L. 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. J. Liang, R. Kumar, Y. Xi, and K. W. Ross, "Pollution in P2P File Sharing Systems," in *INFOCOM*, (Miami, FL), Mar. 2005.
8. J. Chu, K. Labonte, and B. N. 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. S. Saroiu, P. K. Gummadi, and S. D. Gribble, "Measuring and Analyzing the Characteristics of Napster and Gnutella Hosts," *Multimedia Systems Journal* **8**, Nov. 2002.
10. D. Stutzbach and R. Rejaie, "Capturing Accurate Snapshots of the Gnutella Network," in *Global Internet Symposium*, pp. 127–132, (Miami, FL), Mar. 2005.
11. G. D. Forum, "Browse Host Extension." http://www.the-gdf.org/wiki/index.php?title=Browse_Host_Extension.
12. "Gnutella Developer Forum." http://www.the-gdf.org/, 2005.
13. D. Stutzbach and R. Rejaie, "Evaluating the Accuracy of Captured Snapshots by Peer-to-Peer Crawlers," in *Passive and Active Measurement Workshop*, *Extended Abstract*, pp. 353–357, (Boston, MA), Mar. 2005.
14. F. Peers and Inc., "BearShare Network Statistics." http://www.bearshare.com/stats/, Oct. 2005.
15. E. Adar and B. A. Huberman, "Free riding on gnutella," *First Monday* **5**, Oct. 2000.
16. D. Stutzbach and R. Rejaie, "Characterizing Churn in Peer-to-Peer Networks," Tech. Rep. 2005-03, University of Oregon, Eugene, OR, May 2005.
17. N. Leibowitz, M. Ripeanu, and A. Wierzbicki, "Deconstructing the Kazaa Network," in *WIAPP*, 2003.
18. R. Bhagwan, S. Savage, and G. Voelker, "Understanding Availability," in *International Workshop on Peer-to-Peer Systems*, 2003.
19. D. Stutzbach and R. Rejaie, "Characterizing the Two-Tier Gnutella Topology," in *SIGMETRICS*, *Extended Abstract*, (Banff, AB, Canada), June 2005.
20. L. A. Adamic, R. M. Lukose, B. Huberman, and A. R. Puniyani, "Search in Power-Law Networks," *Physical Review E* **64**(46135), 2001.
21. K. P. Gummadi, R. J. Dunn, S. Saroiu, S. D. Gribble, H. M. Levy, and J. Zahorjan, "Measurement, Modeling, and Analysis of a Peer-to-Peer File-Sharing Workload," in *SOSP*, 2003.
22. S. Sen and J. Wang, "Analyzing Peer-To-Peer Traffic Across Large Networks," *IEEE/ACM Transactions on Networking* **12**, pp. 219–232, Apr. 2004.