Spam Characterization and Detection in Peer-to-Peer File-Sharing Systems

Dongmei Jia
Illinois Institute of Technology
10 W 31st Street
Chicago, IL 60616
1-312-567-5330
jiadong@iit.edu

Wai Gen Yee
Illinois Institute of Technology
10 W 31st Street
Chicago, IL 60616
1-312-567-5205
yee@iit.edu

Ophir Frieder
Illinois Institute of Technology
10 W 31st Street
Chicago, IL 60616
1-312-567-5143
ophir@ir.iit.edu

ABSTRACT
Spam is highly pervasive in P2P file-sharing systems and is difficult to detect automatically before actually downloading a file due to the insufficient and biased description of a file returned to a client as a query result. To alleviate this problem, we first characterize spam and spammers in the P2P file-sharing environment and then describe feature-based techniques for automatically detecting spam in P2P query result sets. Experimental results show that the proposed techniques successfully decrease the amount of spam by 9% in the top-200 results and by 92% in the top-20 results.

Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *Search Process*

General Terms
Measurement, Experimentation, Security, Human Factors

Keywords
P2P search, spam, characterization, detection

1. INTRODUCTION
Spam is a well-known problem in P2P file-sharing systems, due to their anonymous, decentralized and dynamic nature [1][2][3][8]. A 2005 study observed that more than 50% of the matching results of popular recent songs were spam [3]. To improve the usability of P2P file-sharing systems, it is important to develop effective spam detection techniques.

We define as spam any file that is deliberately misrepresented or represented in such a way as to circumvent established retrieval and ranking techniques. One reason spam is so prevalent in P2P file-sharing systems is most shared files are not *self-describing*. Shared files are often binary media files that are identified by users by their filenames. A spammer can easily rename a file to manipulate how it is retrieved and ranked. For example, the music/movie industry has been injecting large amounts of spam into the network by naming them after real songs/movies in the battle against the illegal distribution of copyrighted materials [2][3].

Spam is harmful to P2P file-sharing systems in several ways. First, it degrades user experience. Second, spam may contain malware that, when executed, could destroy a computing system. Third, its transfer and discovery waste a significant amount of network and computing resources.

The naïve approach for identifying spam is to download the file and then examine its contents. If the file turns out to be spam, it can be reported on centralized databases (e.g., Bitzi [17]). The obvious problems with this approach are that it consumes time and computing resources and can release malware onto the client.

We propose a way of identifying spam that does not require the download of candidate files. To this end, we:

- Characterize spam
- Characterize spammers
- Propose techniques that use our characterizations to rank query results

Our proposed spam detection also requires little new functionality in existing P2P file-sharing systems. Rather, it relies on captured statistics to detect spam. Our results on Gnutella trace data show that we can decrease the amount of spam by 9% in the top-200 results and by 92% in the top-20 results compared with the base case.

1.1 Outline of the Paper
We discuss preliminaries first. In Section 2, we present related work and contrast it to ours. In Section 3, we specify how queries are processed in P2P file-sharing systems. We describe the four types of spam we have identified in P2P file-sharing systems in Section 4. To identify the spam, we propose in Section 5 the use of query result "features", such as the combinations of terms found in a descriptor and demonstrate how feature values are correlated with spam. In Section 6, we outline a framework for automatic spam detection and present experimental results. We make concluding remarks in Section 7.

2. RELATED WORK
The most widely recognized form of spam is email spam, also known as junk emails, which are unsolicited, nearly identical

CIKM'08, October 26–30, 2008, Napa Valley, California, USA.

messages sent in large quantities to numerous recipients with the purpose of commercial advertising or spreading viruses or other malware. Many popular automated email spam detection methods filter and block email spam by analyzing their content and/or syntax, storing DNS-based blacklists of known spammers' IP addresses or constructing social networks for email addresses. For instance, [4] presents an approach that identifies semantic patterns in emails and then classifies them by applying a back-propagation neural network. [5] proposes MailRank, a social network-based approach, to rank and classify emails according to the address of email senders. It collects data about trusted email addresses from different sources and then creates a graph for the social network via email interactions. However, global information is needed to compute a ranking score for each email address. It is not resilient to attacks from cooperating spammers who build their own social networks. In general, these techniques are not applicable to P2P spam detection because P2P query results are often hard to distinguish (their filenames are relatively short and contain all query terms) and they require global information and the tight integration of users, which is assumed to be infeasible in general P2P environments.

Another well-known type of spam is Web spam – Web pages that are unrelated to the query that appear in search engine results. Many studies have been conducted on detecting Web spam. A variety of methods identify spam pages by analyzing either the content or link structure of Web pages [6][7][25][27]. Again, due to the dynamic and distributed nature of P2P file-sharing systems and the fact that shared files are only represented by small, user-defined file descriptors, Web spam detection techniques are not applicable in the P2P scenario.

The P2P spam detection technique proposed in [3] identifies shared music files as spam if the files are either non-decodable (unplayable) or their lengths are not within +10% or -10% of the official CD version. These techniques require downloading the file and only works for commercial music files whose official CD length is known. Judgments of shared music files encoded in other formats cannot be made. The general idea of using a file size, however, is similar to our feature-based spam detection. Because the file size feature has already been considered, we do not consider it in this work.

[9] takes a different view of the spamming problem in P2P networks. Instead of measuring and detecting spam files, it focuses on the relationship between spamming degree and P2P user behavior (e.g., awareness of spam, elapsed time between download completion and quality checking). Through a controlled study of spam crafted by various content and description manipulation strategies, [9] claims that user awareness is a key factor in pollution dynamics – low awareness of most types of spam and delay on checking the quality of downloaded files result in the unintentional spread of spam. While this may be true, it is orthogonal to the automatic detection of spam.

A spam filter was introduced to LimeWire's Gnutella [10] at the end of 2005. A user can mark a search result that is not relevant to his query or appears to be a virus as junk. Over time, the filter learns from peers that mark search results as junk, and updates the 'rating' of each result accordingly. A result with a high rating is more likely to be considered spam [24]. Compared with this user-controlled approach, our work does not rely on previous user judgments and takes a different approach on automatically detecting spam results.

Several works rely on the experience of other peers with shared files to detect spam without having to download the files. [11][12][16] build reputation systems to allow peers to rank each other, so that peers identified as malicious are less able to share files. However, the success of this mechanism is determined by the honesty level of peers. Instead of judging peers, [13] proposes that individual files be judged by users. The authenticity of a file is evaluated by having the client collect its judgments and evaluate them based on a credibility judgment of the client from which the judgments come. This system requires each peer maintain a vote database for the purpose of vote matching, which may not be scalable in a large system, is resource–intensive, and may be unreliable in environments where peers anonymously join the network for only short periods of time.

3. QUERY PROCESSING SPECIFICATION

In typical P2P file-sharing systems (e.g., Limewire's Gnutella) peers collectively share a set of (binary) files by maintaining local replicas of them. Each replica is represented by a user-tuned descriptor, which includes a filename, some embedded descriptive information (e.g., ID3 data embedded in mp3 files [19]) as well as an identifying *key* (e.g., a SHA-1 hash on the file's bits). All replicas of the same file naturally share the same key. The query processing includes the following major steps:

1. A client issues a query and routes it to all reachable servers until the query's time-to-live expires.
2. A server compares the query to its local replicas' descriptors; a query *matches* a replica if its descriptor contains all of the query's terms. (This is known as "conjunctive" query processing or query matching.)
3. On a match, the server returns its system identifier and the matching replica's descriptor to the client.
4. The client groups individual results by key. Each group is represented by a group descriptor, which is the aggregation of all the result descriptors the group contains.
5. The client ranks each group in the result set by a specific ranking function – generally by the number of results in the group (*group size*).
6. The client becomes a server for the file that is downloaded. The new file is a replica of one of the servers that returned a result in the result set.

4. A CLASSIFICATION OF SPAM

We classify P2P spam to organize our approaches for their detection. Each class of spam is distinct in how their creators attempt to disseminate them. These differences allow us to tailor the various techniques used to detect them.

To classify P2P spam and design and evaluate our spam detection algorithms, we use a collection of "metadata" from 25,137,217 P2P audio files, of which 9,575,113 are unique, shared by 226,786 peers in Gnutella network. The shared data were collected by browsing peers' shared folders using our IR-Wire crawling tool [14] in the Spring of 2007. The information (i.e., metadata) we recorded for each file includes the filenames, unique

identifiers (or *keys*) of files (i.e., SHA1 hash on file's bits), peer identifications (i.e., IP addresses) and file types.

4.1 Classes of P2P Spam

As stated in Section 1, we consider as spam any file that is altered to manipulate the P2P file-sharing system's retrieval or ranking functions. By this definition, a virus file named "spiderman-movie.dvi" is spam, whereas the same file named "virus.exe" is not. Regardless, spammers find ways to place these results at the top of a user's search results.

Spamming is generally performed by manipulating Steps 1, 3 and 5 of the query processing specification stated in Section 3. These steps control who processes the queries, what results are returned to the client and how they are ranked in the result set. They manipulate Step 1 by placing highly active peers in the network that actively participate in file sharing (e.g., see Overpeer [22]). Reputation systems discussed in Section 2 address this problem so it is not in the scope of our work. Rather, we focus on the spam that manipulates query processing Steps 3 and 5, which focus on identifying spam that are in the query result sets sent to clients.

Many of the classes of the P2P spam have analogs in Web spam. These analogs are similar in general approach and we point them out where appropriate.

In our data analyses and our experience using P2P file-sharing systems, we have identified four types of spam.

1. Files whose replicas have semantically different descriptors.

In this case, a spammer tries to disseminate widely a file by replicating it and creating different descriptions for each. A spammer might name a file after a currently popular song. An example of this type of spam found in our crawled audio data set is the file with key 26NZUBS655CC66COLKMWHUVJGUXRPVUF. This file's replicas have descriptors that contain various song titles that refer to several distinct songs, including '12 days after christmas.mp3', 'Niche- Oops Oh My.mp3', 'i want you thalia.mp3' and 'comon be my girl.mp3'. Notice that each of these descriptors looks normal in terms of size and combination of terms. It is only when we compare the descriptors of different replicas does it become clear that the shared file is likely spam.

Type 1 spam has many analogs to Web spam, including Web "content spam" techniques, such as keyword stuffing [25] or Web site scraping [26]. These techniques add popular terms to Web sites to increase their visibility in search results. Type 1 spam is also similar to the Web link spam practice of "page hijacking," where a well-known Web site is copied, but then redirects a user to spam content [27].

2. Files with long descriptors that contain semantically non-sensical terms combinations.

Here, a spammer creates a single file that matches a large class of queries by putting popular terms in their descriptors. This type of spam is different from the first type, as terms in the descriptor together do not attempt to represent an existing file.

For example, the descriptor of a single replica of file key 1200473A4BB17724194C5B9C271F3DC4 is 'Aerosmith,Van Halen,Quiet Riot,Kiss, Poison, Acdc, Accept, Def Leappard, Boney M, Megadeth, Metallica, Offspring, Beastie Boys, Run Dmc, Buckcherry, Salty Dog Remix.mp3.'

Type 2 spam is similar to Type 1 spam with the difference that the additional keywords used to boost the file's ranking are added to a single descriptor instead of being spread out over the descriptors of several replicas.

3. Files with descriptors that contains no query terms.

A server wishing to share a particular file may return the file regardless of whether it matches the user's query. For instance, the result could be advertisements or a warning on the illegality of downloading of copyrighted materials, such as files with the descriptor, "Can you afford 0.09 www.BuyLegalMP3.com.mp3".

Type 3 spam falls under the category of query-independent spam because it manipulates query results independent of the query. On the Web, link spam does the same thing. For example, link farms' goal is to increase the strength of association that of a Web site has with a particular term set [23].

4. Files that are highly replicated on a single peer.

We assume that normal users do not create multiple replicas of the same file on a single server and that the only reason this is done is to manipulate the "group size" ranking technique used on most P2P file-sharing clients or to retard the query routing techniques used to route queries for hard-to-find content [28]. Although the file may be correctly described, because its replication manipulates the ranking function, it is by definition spam.

For instance, in our dataset, all of the 177 replicas of the file with key 6DY2QXX3MYW75SRCWSSUG6GY3FS7N7YC are shared on a single peer.

Type 4 spam is analogous to "duplicate content" Web spam, which aims to increase a site's association with some content terms by duplicating its instances of this content [18][21]. It is also similar to the link-farming technique used by Web spammers to increase a Web site's PageRank.

Among the 4 types of spam, Types 2 and 3 should be easy for any query-dependent similarity-based ranking function (e.g., Cosine similarity ranking) [20] to identify directly and rank low. Hence, they may not be widely spread in the network and would be less harmful. However, their inclusion in query result sets does degrade the user's search experience and wastes network and computing resources.

Types 1 and 4 spam are more difficult to detect because they appear to match the query and be described with a sensible combination of terms. We propose automatic ways of detecting them in this work so as to avoid downloading files from spammers.

5. "FEATURES" OF P2P SPAM

Our approach in the automatic identification of spam files is based on identifying the features that characterize them, such as replication degree, distribution of files over hosts, descriptor lengths, size of the vocabulary used to describe a file and so forth. Ideally, once these features are identified, spam identification becomes a task of feature computation.

To make this task more manageable, we isolate our initial investigations to the files shared by the 50 peers in our data set who share the most files. We pick these peers because their high degree of sharing makes them suspicious. No casual user would share so many files – 8,000 files on average for these peers. In total, the top 50 peers share 401,855 replicas of 149,923 unique

files. The top and bottom peers share 15,844 and 5,452 files respectively.

To the best of our knowledge, no benchmark P2P data set currently exists, esp. the lack of spam judgment, hence, for the purpose of spam evaluation, we manually inspected every unique file and labeled it as spam or non-spam based on the collected file information (e.g., terms in file descriptors) from Gnutella network as well as information retrieved by looking up the file (identified by its key) on Bitzi [17]. For instance, one file in the top 50 peer dataset with different descriptors 'jamie kennedy - matress mack.mp3' 'ball busters prank calls.mp3' and key KVGBBGVZYJ7BFPJBFYIVPAWKNEHMPDKX is labeled as a 27-second audio advertisement of 'efreeclub.com' on Bitzi. Another example of spam in our dataset is file with key QFX3NMHJMOGG7VF7IK5AOFST2L3EWKCA and different filenames such as 'brothers boots when she made me promise.wma', 'earth its easy for u to say.wma' is rated on Bitzi as "Dangerous/Misleading" and one user comment for this file is "Downloaded accidentally - clearly a virus". Among all the 149,923 unique files, 17,129 (11.4%) are labeled as spam.

5.1 Candidate Features

We investigate the correlations between several features of shared files and the peers that share them and prevalence of spam. The features include:

- Replication degree of a file (numRep): We expect that extremely highly replicated files are spam. Such files, with a replication degree beyond that which is expected of an even very popular legitimate file, is suspected of being an attempt to manipulate group size ranking at the client.
- Number of hosts on which a file is shared (numHost): We expect non-spam to be more widely distributed than spam. We expect spammers to be in the minority in the networks and for legitimate users to delete spam if they identify it, so spam will not be distributed widely.
- Average descriptor length of a file (avgDLen): A long average descriptor length may indicate an attempt to match several unrelated queries. This can be measured by numTerms / numRep, where numTerms is the length of file group descriptor.
- Vocabulary size of a file's group descriptor (numUniqueTerms): A file group descriptor is the aggregation of all the replica descriptors of the file. The group descriptor's vocabulary size is the number of unique terms it contains. A larger than normal vocabulary suggests that the file has different descriptions that allow it to match unrelated queries.
- Variance of terms in replica descriptors of a file: High variance in the descriptors of different replicas may indicate an attempt to match several unrelated queries.

  Descriptor variance can be measured by average Jaccard or Cosine distance between replica descriptor D_i and file group descriptor G.

  – Jaccard: The Jaccard distance between a single replica descriptor D_i and file group descriptor G is defined as:

  $$1 - |D_i \cap G| / | D_i \cup G|$$

  Since the term set of replica descriptor D_i is always a subset of the group descriptor G, the second term is equal to $|D_i| / |G|$, which can be interpreted as the ratio of number of terms in replica descriptor to total number of terms in group descriptor, without considering term frequencies. (Term frequency consideration is a major difference between Jaccard and Cosine distance.)

  – Cosine: The cosine distance between replica descriptor D_i and file group descriptor G is defined as:

  $$1 - (V_G \cdot V_{Di}) / (|V_G| \, |V_{Di}|)$$

  where G and D_i are modeled as term frequency vectors (V_G of length $|V_G|$ and V_{Di} of length $|V_{Di}|$). This indicates the degree of dissimilarity between the two term frequency vectors. A high Jaccard or Cosine distance indicates a high variance in the descriptors of different replicas of a file. Notice that Jaccard and Cosine distance only apply to files with multiple replicas (numRep>1).
- Per-host replication degree of a file (repPerHost): This represents how replicas of a file are distributed among peers. We consider a file with a high repPerHost to be abnormally distributed, which may indicate an attempt to manipulate group size ranking on the client. repPerHost is computed as numRep / numHost for a file.
- Average file replication degree on a peer (avgRepDegree): Unlike previous features, this one describes a peer instead of a file. A peer that shares several copies of the same file is likely to be a spammer, as it may intend to manipulate the group size ranking of clients. avgRepDegree is computed as the ratio of total number of files and the number of unique files shared on the peer.

5.2 Effectiveness of Features

Table 1 shows statistics of the various features in the top-50 peer dataset. (More statistics can be found in [29].)

Table 1. Statistics of various P2P features in the top-50 peer dataset

P2P feature	Min	Max	Mean	Median	Std dev
numRep	1	177	2.68	2	2.16
numHost	1	13	1.04	1	0.24
avgDLen	1	46	5.79	5.0	3.13
numUniqueTerms	1	113	5.77	5	3.18
Jaccard	0	0.95	0.02	0	0.07
Cosine	0	0.54	0.01	0	0.02
repPerHost	1	177	2.56	2.0	1.87
avgRepDegree	1	7.68	3.39	3.05	1.71

An indication of the ability of the features to identify spam is how well each can isolate spam when ranking the files shared by the top-50 peers. In Table 2, we show how many of the top 20 files with highest value of each feature are spam. At least 95% of the top 20 files ranked by numUniqueTerms, Jaccard, Cosine or repPerHost are spam, which suggests that they are more strongly

correlated with spam files when compared with numRep, numHost and avgDLen.

Table 2. Percentage of spam in 20 files with highest feature values in the top-50 peer dataset

P2P feature	% spam in top 20 files
numRep	75%
numHost	40%
avgDLen	0%
numUniqueTerms	95%
Jaccard	100%
Cosine	100%
repPerHost	100%

We examine the effectiveness of each feature in identifying spam in more detail in the following sections.

5.2.1 Replication Degree of a File

In this first experiment, we investigate the effectiveness of replication degree (numRep) in identifying spam. In Figure 1, we compare replication degree and likelihood of being spam. Figure 1 also indicates the percentage of files that have particular replication degrees. Most files have a low replication degree, with 36% having only a single replica, 20% having two replicas, and so forth.

The spam-possibility line goes up and down repeatedly and seems to have no clear trend, though there is a decrease when file replication degree increases from 5 to 25. This is evidence for the hypothesis that extremely highly replicated files are spam.

However, the graph does not seem to be a very reliable indicator of spam, given its sudden spikes and valleys. This may be caused by the fact that some popular authentic files are reasonably shared among many peers. Both spam and non-spam can have high numRep values.

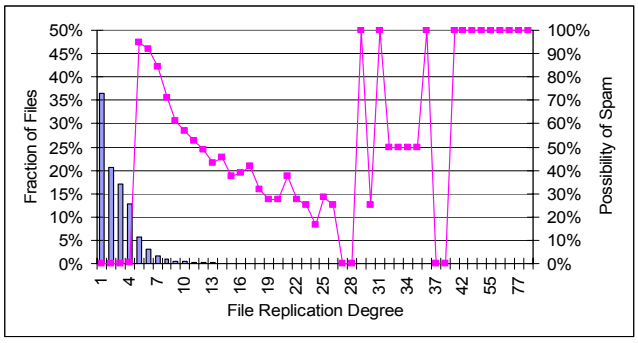

Figure 1. Distribution of spam relative to file replication degree

5.2.2 Number of Hosts Sharing a File

In this experiment, we explore the correlation between the number of hosts who share a file (numHost) and its possibility of being spam. The results in Figure 2 show that 97% (145,567) of files are each found on no more than one peer. Among these files, 90,998 have more than one replica shared on the peer (i.e., numRep>1 and numHost=1). In other words, 95% of the files with multiple replicas in the dataset each have all of their replicas shared on a single peer. This distribution may indicate that there exist potential spammers who share multiple instances of a same file. However, this feature is not a good indicator of spam files, as for almost the entire numHost range, the incidence of spam is lower than 50%. While it is true that the graph monotonically increases to the right, this is due to the fact that there are so few files (fewer than 5 in our data set) replicated on 10 or more hosts.

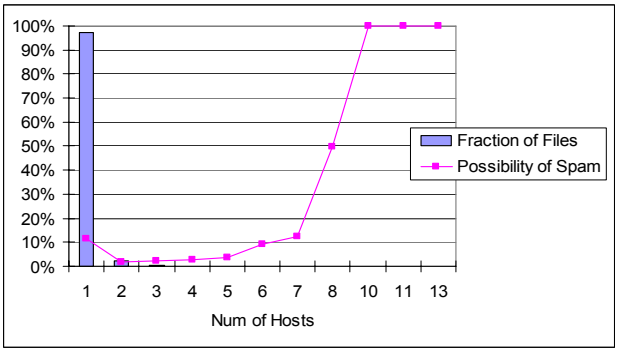

Figure 2. Distribution of spam relative to number of hosts who share a same file

5.2.3 Average Descriptor Length of a File

An examination of the relationship between average length of a replica descriptor (avgDLen) and incidence of spam is shown in Figure 3. Overall, there is no clear upward or downward pattern shown in the spam incidence graph with increasing avgDLen. Furthermore, for almost the entire avgDLen range, the incidence of spam is lower than 50%. Hence we conclude that the correlation between avgDLen and the incidence of spam is low.

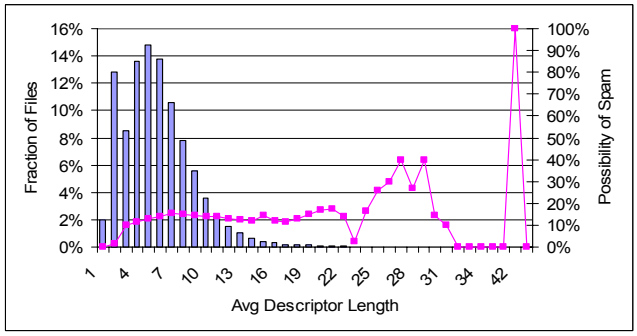

Figure 3. Distribution of spam relative to average file descriptor length

5.2.4 Number of Unique Terms in a File's Group Descriptor

The relationship between the number of unique terms in a file's group descriptor (numUniqueTerms) and the incidence of spam is shown in Figure 4. The incidence of spam increases with numUniqueTerms. The graph gets noisier towards the right due to the variance caused by the small number of files whose descriptors contain many unique terms.

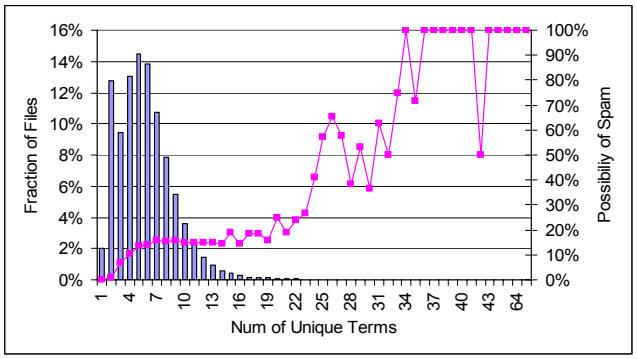

Figure 4. Distribution of spam relative to number of unique terms in file descriptor

5.2.5 Jaccard Distance within Descriptors of a File

We are also interested in how different replicas of a spam file are described. A low Jaccard distance score represents low variance in how different replicas of a file are described. As shown in Figure 5, the incidence of spam increases consistently with Jaccard distance. Hence, files with high Jaccard distance scores (high descriptor variance) are more likely to be spam. Notice that only files that have multiple replicas (numRep>1) are considered in Figure 5 as well as Figure 6, as Jaccard and Cosine distance are not available for file with only one replica.

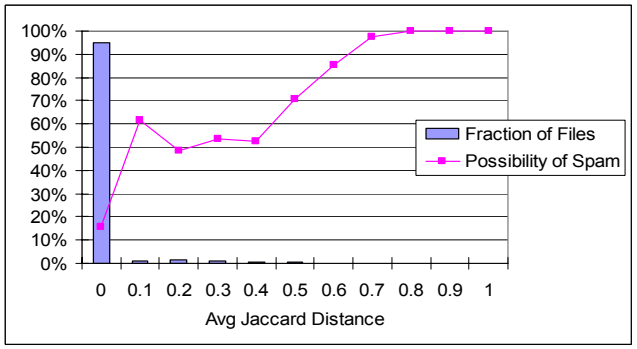

Figure 5. Distribution of spam relative to average Jaccard distance among replica descriptors of a file

5.2.6 Average Cosine Distance within Descriptors of a File

The ability of cosine distance to identify spam is similar to that of Jaccard, as shown in Figure 6. The graph depicting the incidence of spam rises steadily as Cosine distance increases. This suggests that Cosine distance within descriptors of a file may be a good indicator of spam as well.

Because of the similarity of approach between Jaccard and Cosine, we compare the top-20 lists generated by these two features to check for overlap. It turns out that 12 files appear in both lists, indicating that each of the techniques identify different spam.

In the dataset, we observed that, for quite a few spam files, multiple replicas of the spam shared on a same peer are named by appending different number and/or letter combinations to a same filename. For instance, replicas of a spam file with key 6DY2QXX3MYW75SRCWSSUG6GY3FS7N7YC have filenames 'SBB_3F.WAV', 'SBB_41.WAV', 'SBB_1E_0.WAV' and etc. This file has the lowest Jaccard score (0.054) in the dataset. However, its Cosine score is 0.8, which is not the lowest in the dataset. So although Jaccard and Cosine distance are similar, they are not equal.

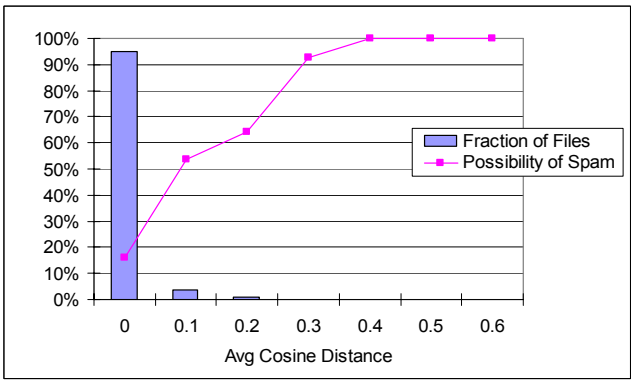

Figure 6. Distribution of spam relative to Cosine distance between replica descriptors and file group descriptor

5.2.7 Per-host Replication Degree of a File

We can use the average replication degree of a file among peers who share it as a heuristic to evaluate the spamming behavior of a peer. Similar to other features, such as numRep, per-host replication degree is designed to identify attempts to manipulate group size ranking. The difference in this case is that the replication degree is normalized by the number of peers that share it, so we avoid the "popular file" problem of numRep (See Section 5.2.1).

As shown in Figure 7 and Table 1, the distribution of files on various per-host file replication degrees is skewed, with a maximum of 177, a mean of 2.56 and a median of 2. Keep in mind that this data is for the top-50 peers, which may not be representative of normal peers in the network. Because their sharing behavior is so different than the average (discussed in more detail below), likely they are spammers. 54,713 files (36%) each have one replica per peer. 17,556 file (11%) have at least 5 replicas per peer. We treat files which per-host replication degree is equal to or greater than 5 as spam in our analysis as a conservative estimate of the impact of P2P spam. We therefore do not indicate in Figure 7 which files are spam; all of the files to the left of numRep=5 are considered spam.

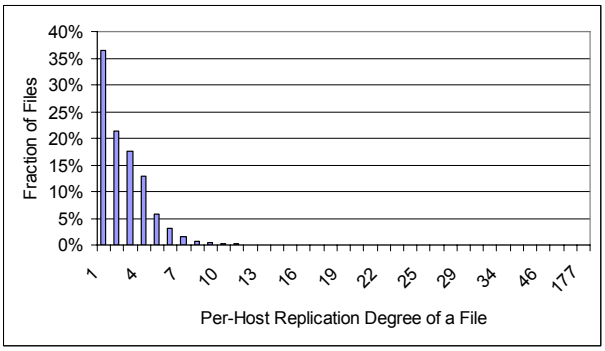

Figure 7. Distribution of files relative to per-host file replication degree

5.2.8 Average File Replication Degree on a Peer

Another way to measure peer "quality" is average replication degree of all the files shared on a peer. In this experiment, we

compute the average replication degree of the collections of each of the top-50 peers and examine the correlation between this peer feature and percentage of spam shared by the peer. In Figure 8, as expected, peers with high file replication degrees share more spam. This indicates that such peers are very likely spammers.

To understand better how differently spammers and normal peers behave in terms of sharing multiple replicas of a same file, we created another 50-peer dataset by random selection of peers. In this random-50 peer dataset, the total number of files is 30,444, of which 24,932 are unique.

As shown in Figure 9, by comparison, the file replication degree on the random 50 peers is much lower than that of the top 50 peers. Most of the 50 random peers share only a single copy of each file, which reinforces the statement that a normal peer shares only one copy of a file in general. To be exact, only 2 peers share two or more copies per file on average among the 50 random peers, whereas this number is 39 in the top 50 peer dataset. Hence, average file replication degree on a peer is a good indicator of a spammer.

More evidence of the suspicious behavior of the spammers can be observed by analyzing the degree of commonality amongst their collections and the degree of commonality between their collections and that of a random peer. We visualized this evidence via a graph of the top-50 peers and the 50 random peers shown in Figure 10. Each node represents a peer and the size of a node corresponds to the number of files shared on the peer. Peers with file replication degree equal to 1 and greater than 1 are colored in black and white, respectively. An edge is drawn between two nodes if the two peers share at least 30 files in common.

We observe that most of the white nodes cluster to form a single connected graph with a radius less than 10. Most of the black nodes, on the other hand, are isolated with no connections to others. (We excluded these nodes from the figure to simplify it.) This is strong evidence that suspected spammers cooperate with each other. However, average peers share files based on their unique interest and therefore have little linkages to randomly selected peers. Finally, the lack of edges between the black and white nodes – despite the white nodes' linkages to each other – indicates the segregation. This is natural if the white nodes shared spam and the black nodes were normal users; a normal user may download spam, but would delete it as soon as it is identified as spam.

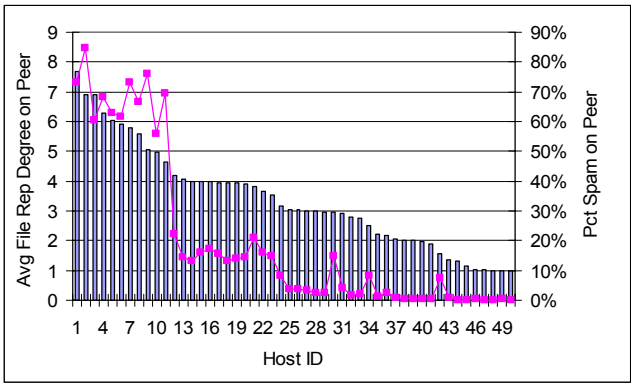

Figure 8. File replication degree and percentage of spam on top 50 peers

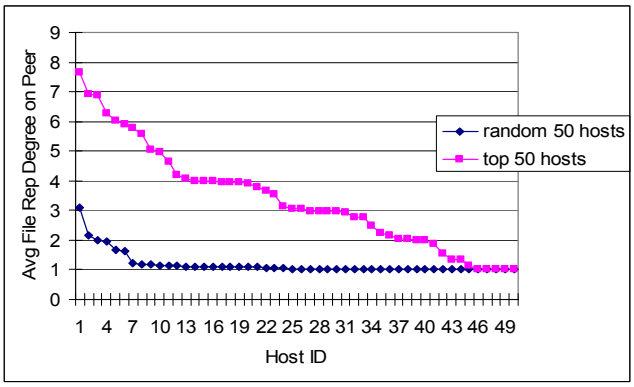

Figure 9. Comparison of file replication degree on peer between top-50 peers and random-50 peers

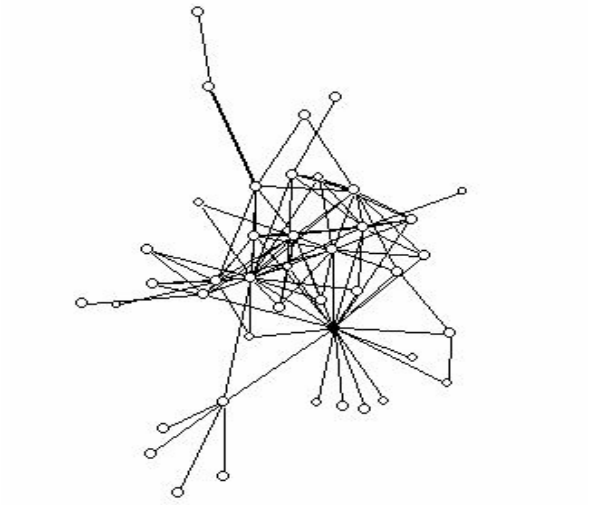

Figure 10. Node graph for peers with various file replication degrees

To summarize, features such as number of replicas, number of hosts and average descriptor length of a file are not strongly correlated with the possibility of spam. However, vocabulary size, variance of replica descriptors, and per-host replication degree of a file are good indicators of spam; replication degree of a peer's shared content is a good indicator of spammer. In the next section, we explain how to use these features to detect spam.

6. FEATURE-BASED SPAM DETECTION

Our goal is to improve P2P search accuracy by automatically identifying spam in P2P query result sets without requiring a user to download any files. We take the basic query processing steps outlined in Section 3 and make modifications to them to accommodate spam detection.

6.1 Detecting Types 2 and 3 Spam

Current P2P clients rank search results typically based on server or file quality such as available bandwidth or relative popularity of the file (i.e., group size). Recent work demonstrates that randomly ranking search results is substantially more reliable than these ranking functions [15].

Group size ranking is not effective on identifying spam results, as the number of copies of a spam file may exceed the number of an authentic file in result set. Our experiments on the numRep feature (see Table 2 and Figure 1), which show the unreliability of replication degree in identifying spam, is more evidence of the failures of group size ranking in this regard.

To identify Types 2 and 3 spam, we propose a straight-forward application of query-dependent IR-ranking techniques. Types 2 and 3 spam are characterized by several terms that are irrelevant to the user's query. Group size ranking, by being query independent, does not identify such spam. However, query-dependent ranking functions [20], such as Cosine similarity or Okapi BM25 naturally identify Types 2 and 3 spam.

The modification we make to the steps of query processing to detect Types 2 and 3 spam is replacing step 5 with the following:

5a. Groups are ranked by cosine similarity (or some other query-dependent ranking function).

6.2 Detecting Types 1 and 4 Spam

Types 1 and 4 spam are not detectable with query-dependent because they naturally resemble the query. In this case, we make use of the features identified above to identify spam.

Recall that Type 1 spam is characterized by variance among the descriptors of its replicas. This type of spam is identifiable by the following two features:

- Vocabulary size of a group (numUniqueTerms).
- Variance of replica descriptors of a group (Jaccard or Cosine distance).

Type 4 spam is characterized by its high replication degree per peer. This type of spam is identified by the following feature:

- Per-host file replication degree (repPerHost).

To integrate these features into the query processing steps, we propose the following steps after Step 5a and before Step 6:

5b. Identify the top-*M* results as *candidate* results.
5c. Re-rank the top-*M* results by either NumUniqueTerms or Jaccard/Cosine distance. The results that are low in the order are more likely to be Type 1 spam than those higher up.
5d. Identify the top-*N* results, where $N < M$ as the new *candidate* results.
5e. Re-rank the top-*N* results by their per-host file replication degree. The results that are low in the order are more likely to be Type 4 spam than those higher up.

These steps isolate candidate results from the first ranking and re-rank them to identify Type 1 spam in within the candidates (Steps 5b and 5c). We repeat this process for Type 4 spam (Steps 5d and 5e).

6.3 Probe Queries to Enhance Spam Detection

One of the challenges in detecting spam is that the query results will tend to look alike due to the conjunctive matching condition. For example, one of our proposed methods for detecting Type 1 spam is to identify variance among the replicas' descriptors. Yet, conjunctive matching only retrieves the replicas of a file with descriptors that resemble the query (and therefore resemble each other), while not retrieving replicas of the same file with very different descriptors.

To solve this problem created by conjunctive matching, we propose the use of "probe queries," [30] which, given a file's key, searches for its feature information relevant to spam detection from other peers in the network.

A probe query contains only the key of a result file and is sent to peers who share this file in the network. A peer responding to a probe query sends back local descriptor(s) of the probed file, the total number of replicas, the number of unique files shared on this peer and the identifier of the peer.

By issuing a probe query for a file, we create a more complete view of how a file is shared. This information (e.g., descriptors that do not match the original query, servers who act like spammers) is used to identify it as spam.

To integrate probing into the query processing steps described above, we insert the following step after Step 5b:

5b'. Issue probe queries for the top-*M* results.

The information collected from the probe query issued for the candidate results will help in determining whether they are Types 1 or 4 spam.

6.4 Simulating P2P Search

To evaluate the effectiveness of the proposed techniques, we simulate a P2P search on a client's perspective using the data we crawled from Gnutella network. On these data, we issue the top 50 most popular queries for audio files that we identified from our crawled data. We use these queries as they are representative of the most users and likely targets for spam.

The client issues the basic 6 steps outlined above for query processing, with variations based on the experiment. To simulate P2P query routing, without loss of generality, a query is randomly sent to a given number of peers (i.e., 50 peers) who return matching results. This process repeats until the number of results returned to the client reaches a given threshold (i.e., 200 results) or a threshold number of peers have received the query (i.e., 50,000 peers). Threshold values were chosen based on the specifications of a real-world P2P file-sharing system (e.g., LimeWire's Gnutella [10]).

We manually judge the retrieved results for each of the 50 queries as spam or non-spam based on the description of spam Type 1 to 4 as introduced in Section 4.1. Performance is measured using a standard metric – the number of the top *N* ranked results that is spam, especially when *N* is small, as a user tends to look at only a few top-ranked results.

6.5 Experimental Results

To test the proposed probing and ranking techniques, we compare them with the two no probing (noprobe) base cases where group size (numRep) ranking and Cosine similarity (CosineQD) ranking are performed.

As discussed earlier, spam Types 2 and 3 (i.e., a file containing many random noisy words in a replica descriptor) are identified by any query-dependent, content-based similarity ranking such as

Cosine similarity even with no probing, as terms in this type of spam's descriptors are irrelevant to query. Spam Type 4 (i.e., a file highly replicated on a single peer) is detected by the proposed ranking – per-host file replication degree (repPerHost), which can be easily computed based on the statistics (i.e., number of replicas of a file, number of peers who share a file) obtained by probing. Hence, in our experiments, we focus on examining how the proposed content-based file "quality" ranking functions with the assist of probing perform on the detection of Type 1 spam.

Figure 11 presents the average amount of spam in the top *N* result sets of 35 of the 50 queries. (Fifteen queries returned no spam, so they are excluded from the performance analysis.) Compared with the two no-probing base cases, the proposed probing-based ranking functions (Cosine, Jaccard and numUniqueTerms) are better at ranking spam low in result set, especially when the value of top *N* is small. For instance, compared with the base case, noprobe+numRep, probe+Cosine improves the performance by 9% over all results and by 92.5% for the top-20 results. Compared with the base case noprobe+CosineQD, the two numbers are 21.6% and 97.8% respectively.

We also examine how numRep performs in the case of probing. The results show that probe+numRep performs the worst, which suggests that group size has trouble detecting spam results, especially when such a spam file is widely spread in the network.

As shown in Figure 11, the ranking function numUniqueTerms seems to perform better than Cosine and Jaccard when the value of top *N* is larger. The reason for this is Cosine and Jaccard distance can only be computed for files with multiple replicas; however, the number of unique terms of a file can be computed even if there is only a single replica of a file.

In order to compare Cosine, Jaccard with numUniqueTerms in a fair way, we consider only multi-replica result files in ranking, and recomputed the average number of spam in top-N results. As shown in Figure 12, Cosine and Jaccard performs consistently better than numUniqueTerms in the case of multi-replica files.

Probing on query results may introduce extra network cost. However, we argue that, compared with the cost on downloading large media spam due to user's unawareness, it is worth to apply probing to filter spam out in advance. Furthermore, because probe queries are only issued to the top-ranked results, the cost should not be increased dramatically.

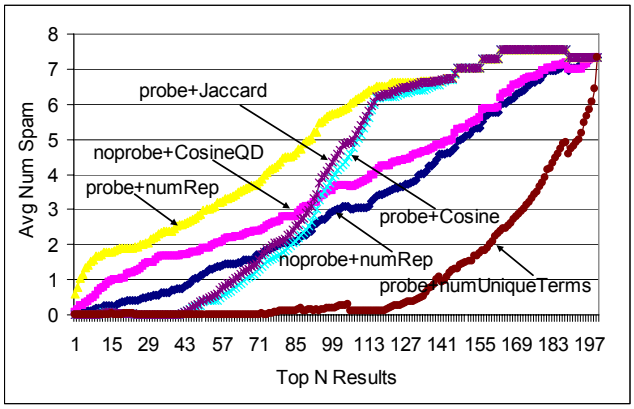

Figure 11. Number of spam in top-N results with various ranking and probing techniques

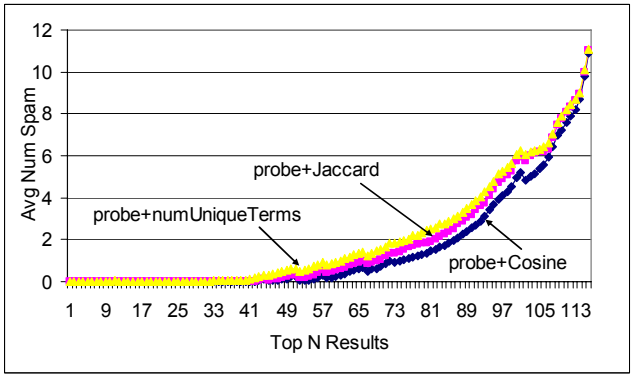

Figure 12. Number of spam in top-N results with various ranking functions (only multi-replica files are considered)

7. CONCLUSIONS

Insufficient and biased description of a file returned to a client as a query result makes it difficult to detect spam automatically before actually downloading a file. By characterizing spam in the P2P file-sharing environment, we reveal that P2P features such as vocabulary size, variance of replica descriptors, and per-host replication degree of a file are strongly correlated with the possibility of spam; replication degree of a peer's content is a good indicator of spammer. Then we propose probing technique that aggregate more descriptive information of result files and statistics of peers and ranking functions that use our characterizations to rank query results. The experimental results demonstrate that the proposed techniques improve the ability to detect spam by 92.5% over the top 20 results.

Because our work requires little new functionality in existing P2P file-sharing systems, it can be combined easily with other existing techniques discussed in Section 2 (e.g., using file size, social feedback) to detect more types of P2P spam.

To boost accuracy, we are currently working on ways of combining features into single "spam probability" scores. We are also working on characterizing peers based on an analysis of their collections to determine if they are spammers and identifying other possible types of P2P spam. To reduce cost, we are now exploring ways of limiting the scope of the probing process.

8. ACKNOWLEDGMENTS

We thank Evan Estola and Jason Soo in Information Retrieval Lab at IIT for their help on spam evaluation.

9. REFERENCES

[1] S. Shin, J. Jung, H. Balakrishnan. Malware Prevalence in the KaZaA File-Sharing. Network. *In Proc. of the Internet Measurement Conference (IMC)*, ACM 2006.

[2] N. Christin, A. S. Weigend and J. Chuang. Content Availability, Pollution and Poisoning in Peer-to-Peer File Sharing Networks. I*n ACM E-Commerce Conference (EC'05)*, June 2005.

[3] J. Liang, R. Kumar, Y. Xi and K. Ross. Pollution in P2P File Sharing Systems. *In Proc. of INFOCOM'05*, May 2005.

[4] R. Hashemi, M. Bahar, K. D. Tift, and H. Nguyen. Spam Detection: A Syntax and Semantic-based Approach. *In proc.*

International Conf. on Information and Knowledge Engineering (IKE'06), Las Vegas, Nevada, June 2006.

[5] P. A. Chirita, J. Diederich, and W. Nejdl. MailRank: Using ranking for spam detection. *In proc. CIKM'05*, Bremen, Germany, 2005.

[6] Qingqing Gan and Torsten Suel. Improving Web Spam Classifiers Using Link Structure. *In Third International Workshop on Adversarial Information Retrieval on the Web (AIRWeb'07)*, Banff, AB, Canada, May 2007.

[7] A. Ntoulas, M. Najork, M. Manasse, D. Fetterly. Detecting spam web pages through content analysis. *In Proc. of WWW'06*.

[8] J. Liang, N. Naoumov, K. Ross. The Index Poisoning Attack in P2P File Sharing Systems. *In proc. of INFOCOM*, Barcelona, Spain, Apr. 2006

[9] Uichin Lee, Min Choi, Junghoo Cho, Medy. Y. Sanadidi, Mario Gerla. Understanding Pollution Dynamics in P2P File Sharing. *In Proc. IPTPS'06*.

[10] Limewire. www.limewire.org

[11] D. Dutta, A. Goel, R. Govindan, H. Zhang, The Design of A Distributed Rating Scheme for Peer-to-peer Systems, *In Proc. of Workshop on the Economics of Peer-to-Peer Systems*, 2003

[12] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The EigenTrust Algorithm for Reputation Management in P2P Networks. *In Proc. of the Twelfth International World Wide Web (WWW) Conference*, May, 2003.

[13] Kevin Walsh, Emin Gun Sirer. Experience with an Object Reputation System for Peer-to-Peer Filesharing. *In 3rd Symposium on Networked Systems Design & Implementation (NSDI)*, 2006.

[14] L. T. Nguyen, W. G. Yee, D. Jia, and O. Frieder, A Tool for Information Retrieval Research in Peer-to-Peer File Sharing Systems, *In Proc. IEEE ICDE*, 2007.

[15] D. Dumitriu, E. Knightly, A. Kuzmanovic, I. Stoica and W. Zwaenepoel. Denial-of-Service Resilience in Peer-to-Peer File Sharing Systems. *In Proc. Of ACM SIGMETRICS'05*, Banff, AB, Canada, June 2005.

[16] Runfang Zhou and Kai Hwang. Gossip-based Reputation Aggregation for Unstructured Peer-to-Peer Networks. *21th IEEE International Parallel & Distributed Processing Symposium (IPDPS'07)*, Los Angeles, March 26-30, 2007

[17] Bitzi website. www.Bitzi.com

[18] Google Duplicate Content Web Site. http://www.google.com/support/webmasters/bin/answer.py?answer=66359. Accessed May 25, 2008.

[19] M. Nilsson. Id3v2 web site. www.id3.org.

[20] D. Grossman and O. Frieder. Information Retrieval: Algorithms and Heuristics. Springer, second edition, 2004.

[21] Steve Webb, J. Caverlee, and C. Pu. Characterizing Web Spam Using Content and HTTP Session Analysis. In *Proc. 4th Conf. on Email and Anti-Spam (CEAS)*, 2007.

[22] J. Macguire. Hitting P2P Users Where It Hurts, In Wired, Jan. 13, 2003. http://www.wired.com/entertainment/music/news/2003/01/57112

[23] Googlebombing 'failure.' Official Google Blog. Sept. 16, 2005. http://googleblog.blogspot.com/2005/09/googlebombing-failure.html

[24] http://wiki.limewire.org/index.php?title=Junk_Filter

[25] K. Svore, Q. Wu, C.J.C. Burges and A. Raman. Improving Web spam classification using Rank-time features. In Proc. *AIRWeb workshop in WWW*, 2007

[26] http://en.wikipedia.org/wiki/Web_scraping#References

[27] J. Caverlee and L. Liu. Countering Web Spam with Credibility-Based Link Analysis. *In Proc. the 26th ACM Symposium on Principles of Distributed Computing (PODC)*, 2007.

[28] The Gnutella protocol specification v0.6. http://rfc-gnutella.sourceforge.net.

[29] W. G. Yee, L. T. Nguyen, O. Frieder. A View of the Data on P2P File-sharing Systems. In Proc. Wkshp. Large Scale Distributed Systems for Inf. Retr. (LSDS-IR), 2007.

[30] D. Jia, W. G. Yee, L. T. Nguyen, O. Frieder. Distributed, Automatic File Description Tuning in P2P File-Sharing Systems. *Springer Journal of Peer-to-Peer Networking and Applications*, 2008.