Relating Query Popularity and File Replication in the Gnutella Peer-to-Peer Network

Alexander Klemm, Christoph Lindemann, and Oliver P. Waldhorst
University of Dortmund
Department of Computer Science
August-Schmidt-Str. 12
44227 Dortmund, Germany
http://www4.cs.uni-dortmund.de/~Lindemann/

Abstract

In this paper, we characterize the user behavior in a peer-to-peer (P2P) file sharing network. Our characterization is based on the results of an extensive passive measurement study of the messages exchanged in the Gnutella P2P file sharing system. Using the data recorded during this measurement study, we analyze which queries a user issues and which files a user shares. The investigation of users queries leads to the characterization of query popularity. Furthermore, the analysis of the files shared by the users leads to a characterization of file replication. As major contribution, we relate query popularity and file replication by an analytical formula characterizing the matching of files to queries. The analytical formula defines a matching probability for each pair of query and file, which depends on the rank of the query with respect to query popularity, but is independent of the rank of the file with respect to file replication. We validate this model by conducting a detailed simulation study of a Gnutella-style overlay network and comparing simulation results to the results obtained from the measurement.

Keywords:
Traffic measurement and characterization,
characterization of user behavior,
peer-to-peer systems.

1 Introduction

Peer-to-peer (P2P) systems constitute one of the most popular applications in the Internet. Browsing through the list of applications that are build on Sun's JXTA protocol suit for P2P communication [5] reveals that P2P technology is employed for instant messaging, web publishing, distributed data management, gaming, and many other applications. Nevertheless, most people associate P2P applications with file sharing solutions, which were made popular by Napster [9] and Gnutella [6]. While Napster has gone out of service because of legal troubles, the growth of Gnutella and Gnutella-like systems continues. Since Gnutella's release by AOL affiliate Nullsoft in 2000, many weaknesses in the original protocol design motivated research projects around the world. As all P2P file sharing systems, Gnutella consists of two building blocks: (1) a search algorithm for transmitting queries and search results and (2) a file transfer protocol for downloading files matching a query. While most file sharing systems transfer files between peers using direct TCP connections, efficient searching in P2P systems is an active area of research. Possible approaches to searching include variants of the unstructured search algorithm used in Gnutella, e.g., as employed by Morpheus [15] and KaZaA [12], and structured approaches based on distributed hash table systems, e.g., as CAN [10] and CHORD [13]. Unstructured systems do not provide a coupling between data and location so that a query must be sent to many peers. In contrast, structured systems improve search efficiency by positioning data at exactly those locations to which a query for this data is routed. Recent approaches even propose data replication at multiple locations to improve searching in unstructured networks [4].

Designing a search protocol for a P2P file sharing system, regardless if structured, unstructured, or following any other approach, requires the evaluation of different design alternatives. In early stages of the design process, analytical models can support design decision by providing aggregate measures of protocol performance. However, later design stages require detailed simulation studies or even field studies based on software prototypes. Such performance evaluations need both a detailed model of the considered system as well as a detailed workload model to mimic the load that the system has to bear during operation. An important aspect of a detailed workload model constitutes a user's active behavior, i.e., the generation of queries, and it's passive behavior, i.e., sharing files and responding to queries of other peers.

In this paper, we characterize user behavior in a P2P file sharing network by answering the questions "Which queries do users issue?" and "Which files do users share?" To answer these questions, we use the results of a passive measurement study of the Gnutella file-sharing network. Based on the results of this study, we characterize both query popularity and file replication. As a major contribution, our paper closes the loop between these two measures by introducing an analytical formula describing the matching of files to queries. This matching can be characterized by a probability function that depends on a query's rank with respect to query popularity, but is independent on a file's rank with respect to file replication. To illustrate the accuracy of the proposed workload model, we conduct a simulation study of a Gnutella-style file-sharing network that employs our workload model. The study illustrates that the load experienced by a peer in the simulated network closely matches the load recorded in the traces measured in the Gnutella network.

The remainder of this paper is organized as follows. Section 2 summarizes related work in measurement and workload modeling of P2P file sharing systems. In Section 3, we present the results of a detailed measurement study in the Gnutella network, which are used to characterize query popularity and file replication. Based on the measurement results, Section 4 characterizes the relationship between these two measures by presenting an analytical formula describing the matching probability of files to queries. The formula is validated in a simulation study in Section 5. Finally, concluding remarks are given.

2 Related Work

Several workload studies of P2P file sharing systems have been presented. Sripanidkulchai [14] analyzed the popularity of queries in the Gnutella network. He showed that the popularity of Gnutella queries follows a Zipf-like distribution and proofs that caching of query results can reduce the network traffic up to a factor of 3.7. Sariou, Gummadi, and Gribble [11] performed measurements in the Napster and Gnutella file sharing systems in order to characterize the peers in terms of bottleneck bandwidth, network latency, session duration, number of shared files and number of downloads. They identified different classes of peers and argued that different tasks in a P2P file sharing system should be delegated to different peers depending on their capabilities. Again, using measurements in the Gnutella network, Adar, and Hubermann [1] discovered a significant amount of free riders, which download files from other peers without sharing any files. They argued that free riding degrades the system performance and, therefore, proposed to incorporate mechanism to minimize free riding in future file sharing systems. A comprehensive analysis of locality in shared files and downloads is provided by Chu, Labonte, and Levine [3]. They periodically collected shared file lists from Napster and Gnutella clients over a period of several weeks. The analysis of this data showed that both file locality as well as download locality fit to a log-quadratic distribution.

The measurement study presented in this paper relates a peer's active and passive behavior by characterizing the relationship between query popularity and file replication. Therefore, it builds upon some aspects already known from these previous works. Similar to [14], we characterize peer's active behavior by investigating query popularity. Similar to [1], [3], [11], we characterize the passive behavior of a peer by the amount of file replication. Beyond [1], [3], [11], [14], we characterize the matching between files and queries, closing the loop between a peer's active and passive behavior.

First approaches to modeling the performance of entire P2P file sharing systems constitute [7] and [16]. Yang and Garcia-Molina presented in [16] an analytical model for hybrid P2P systems and evaluated several approaches in terms of the number of query results, CPU and memory requirements. For the validation of the model they used aggregated measures obtained from the server of a hybrid P2P system. Ge, Figueiredo, Jaswal, Kurose, and Towsley presented an analytical model based on a closed multi-class queuing network, which can be tailored to different file sharing systems by appropriately choosing model parameters [7]. Using this model, they analyzed the throughput of file transfers for different types of file sharing systems and user behaviors. The workload used in their model is inspired by the measurement studies presented in [1], [3], and [11].

Previous papers [7] and [16] both provide performance models of an entire P2P file sharing system, focusing on system design and network environment. Client behavior is not incorporated for each individual peer, but aggregated based on the structure and mechanisms of the underlying P2P file sharing system. Thus, none of the models can be employed for detailed simulation studies or prototype-based evaluations. Opposed to [7] and [16], this paper characterizes the relationship of query popularity and file replication as building blocks of a detailed simulation model.

3 Measurement and Characterization of P2P Workloads

3.1 Measurement Methodology for the Gnutella Network

As building block for the presented workload model, we conduct passive measurements in the popular Gnutella overlay network [6]. Since the Gnutella protocol specification is publicly available, the overlay network built by this protocol specification is used by a series of client programs, including commercial products as Morpheus [15]. The Gnutella protocol specifies four message types, two for building and maintaining the overlay network, and two for transferring keyword-based queries and query results. Messages of types *PING* and *PONG* are used to maintain overlay connectivity and obtain information about other peers. Messages of type *QUERY* contain a query-string, i.e., a set of keywords from the title of files a user wants to download. These query messages are transferred to other peers by flooding the overlay network. If a peer shares files, which match to the query string in a query message, it responds with a message of type *QUERYHIT*. This response message is transferred to the inquiring peer on the reverse overlay path the query message was routed to the responding peer.

To perform the measurements in the Gnutella network, we modify the open-source Gnutella client *mutella* [8] to trace the data contained in Gnutella messages originated at remote peers. We conduct only passive measurements, i.e., we do not generate messages actively, minimizing the disturbance of the actual network traffic by the measurement. To maximize the number of messages in the traces, we modify the measurement client to maintain up to 200 connections to other peers.

To illustrate the measurement setup, Figure 1 shows a part of the Gnutella overlay network with six peers and the measurement peer. In this figure, Gnutella peers are shown as circles. Solid lines between peers represent connections in the overlay network. Dashed arrows denote transmissions of QUERY messages and dotted arrows denote transmissions of QUERYHIT messages. In this example peer 1 sends a QUERY message matching to documents shared by peers 5 and 6. The QUERY message is flooded through the overlay network and, subsequently, reaches all peers including the measurement peer. However, the QUERYHIT messages are transferred on the reverse path on which the responding peer received the QUERY message. Therefore, the QUERYHIT message sent by peer 5 does not traverse the measurement peer. Thus, not all QUERYHIT messages can be traced using this measurement setup. We illustrate the impact of this inaccuracy in the following sections.

Consistent with [1], [11], messages containing private IP addresses due to native address translation (NAT) are discarded, because many peers in the Gnutella network may use the same private IP address. These peers cannot be distinguished and would cause errors in the measurement results. The measurement client traced the Gnutella traffic over three periods, September 22, 2003 to October 07 2003, November 11, 2003 to November 26, 2003, and January 09, 2004 to January 29, 2004, providing an overall trace of 53 days. We brake down the measured data into shorter time intervals of individual day times as typically done in measurement studies of Web servers, e.g., [2]. We find that all important characteristics of our workload model are independent of time-of-day or day-of-week. Thus, we will derive most measures from a sub-trace of three days length rather than from shorter sub-traces. Note that the memory requirements for analyzing the trace prohibit the analysis of longer sub-traces. A more detailed discussion on the choice of the analyzed sub-trace is given in Section 3.5.

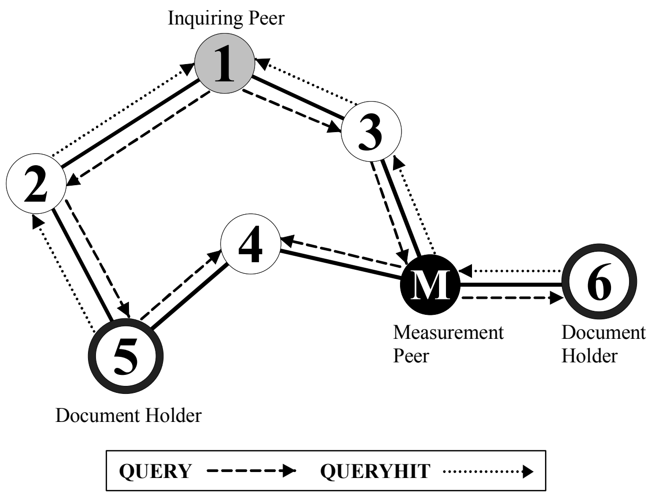

Figure 1. Measurement setup and message routing in the Gnutella overlay network

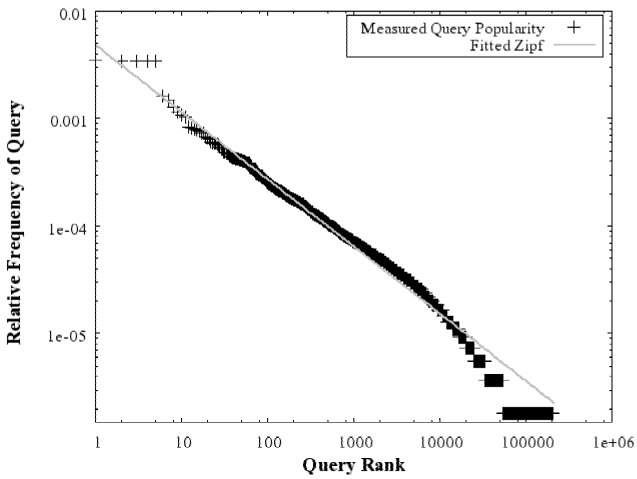

Figure 2. Query popularity and fitted Zipf-like distribution with parameter γ=0.55

3.2 Measuring Query Popularity

We characterize the peers' query behavior by analyzing the query *popularity*, i.e., the frequency of a particular query string in all query messages. To identify identical query messages, we use the following simple heuristic: The query strings are split into keywords at delimiter characters. Two query messages are assumed identical, if the sets of keywords are equal. Figure 2 shows a log-log plot of the relative frequency of queries over the queries' ranks with respect to popularity. Consistent with the measurements conducted in [14], our measurements show the linear shape of a Zipf-like distribution in the log-log plot. The fitted parameter has the value γ=0.55.

3.3 Measuring File Replication

Another important measure for characterizing P2P file sharing systems is the number of copies (*replicates*) of a specific file existing in the system. For this measure, we identify identical files by the file name and file size. We use the following heuristic to identify

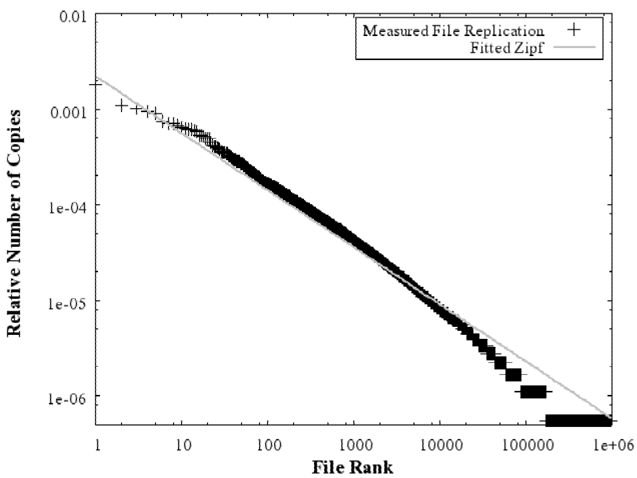

Figure 3. Replication of files and fitted Zipf-like distribution with parameter α=0.60

identical files: A set of words is extracted from the file name by splitting the name at delimiter characters. File names are assumed to be equal, if the word sets of the file names are equal. Files are considered identical, if the file names are equal according to this heuristic, and file sizes are equal, too. Figure 3 plots the relative replication of a file, i.e., the number of copies divided by the sum of files shared by all peers, versus the rank of the file on a log-log scale. The graph shows an approximately linear shape, indicating again a Zipf-like distribution with the parameter α=0.60.

3.4 Measuring the matching between Files and Queries

Recall that the active behavior of a peer is determined by the queries it sends out, while the shared files, for which the peer generates responses, determine the passive behavior. To close the loop between active and passive behavior, queries must be related to the shared files by defining a matching. In a passive measurement study, the files matching to a query can be determined by recording the files reported in QUERYHIT messages to the query. Recall that due to the structure of the Gnutella overlay network, one cannot assure that all response messages to a query message are received. However, using a sufficiently large trace maximizes the probability that QUERYHIT messages for each file matching to a specific query are received at least once.

In our observation period of 53 days, we record query messages for N=211,361 unique queries and response messages for M=4,492,771 unique files. By matching files in response messages to queries, we are able to generate an index that maps each query to all matching files. Formally, we can determine the set $\mathbb{I}=\left\{\left(q_n,\mathbb{F}_{q_n}\right)\right\}$, $1\le n\le N$, where $\mathbb{F}_{q_n}$ denotes the files matching to query q_n. It holds $f_m\in\mathbb{F}_{q_n}$ for $1\le n\le N$, $1\le m\le M$ if and only if f_m was found in at least one response message that was sent in reply to a query message with query q_n. Consider for example a query with string q_n="Madonna Girl". Then, the set of matching files could be given by $\mathbb{F}_{q_n}$={"Madonna – Who's that girl.mp3", "Madonna – Material girl.mp3"}. Note that $\mathbb{I}$ is deterministic for our observation of 53 days. We will show how to use $\mathbb{I}$ to construct the matching of files to queries for shorter simulation runs in Section 4.

3.5 General Applicability of Results

In the following, we discuss why the workload characterization presented in the previous sections is representative and, thus, generally applicable. As stated above the measured trace represents 53 days in three periods. This large data set builds the foundation for deriving a representative workload characterization as summarized by Table 1. To verify that the characterization of the considered sub-trace of three days length (Jan 24 – 26) is representative we characterize a number of sub-traces of three days and one-day lengths within the overall trace. The comparison of the results shows that the probability distributions

Workload measure	Fitted distribution	Probability density function	Matched parameters			
			3-days trace Nov 11-13	3-days trace Jan 24-26	1-day trace Nov 21	1-day trace Jan 19
Query popularity	Zipf-like	$p(r) \cong r^{-\gamma}$	$\gamma = 0.65$	$\gamma = 0.55$	$\gamma = 0.60$	$\gamma = 0.54$
Number of replicates per file	Zipf-like	$p(r) \cong r^{-\alpha}$	$\alpha = 0.53$	$\alpha = 0.60$	$\alpha = 0.53$	$\alpha = 0.58$

Table 1. Distributional models and fitted parameters for different sub-traces.

for each measure are identical for all sub-traces. However, especially for the measures representing the active peer behavior the parameter sets of the fitted distributions vary for different sub-traces. The parameter sets of the fitted probability distributions for two sub-traces of three days length and two sub-traces of one-day length are presented in Table 1, too. It shows that the parameters of both the query popularity distribution and the file replication distribution are quite stable across the sub-traces. Therefore, our further calculations are based on the parameter set derived from the three days trace of Jan 24 – 26, highlighted in gray in Table 1.

4 Relating Query Popularity to File Replication

For investigating the relationship between query popularity and file replication we consider a finite time period of length *T*. In this period, a peer will issue a finite number of queries and shares a finite number of files. Thus, the overall sums of the numbers of unique files and queries, respectively, are finite, too. We denote the overall set of unique queries as $\mathbb{Q}$ and the overall number of unique queries as $N := |\mathbb{Q}|$. Recall that queries are not equally popular, as shown in Section 3. We identify queries by their rank with respect to popularity, i.e., the query 1 is the most popular query and the query *N* is the least popular one. This allows an enumeration $\mathbb{Q} = \{1, 2, ..., N\}$. Furthermore, we denote the set of unique files as $\mathbb{F}$ and the number of unique files as $M := |\mathbb{F}|$. As queries, files are not equally

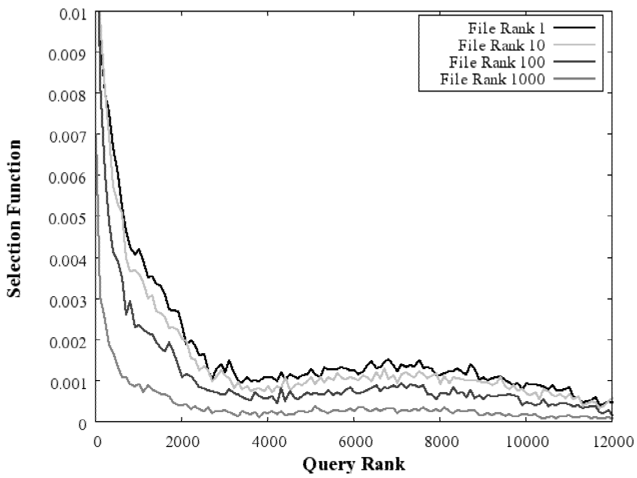

Figure 4. Measured selection function for various file ranks

popular. In fact, some files are more often replicated than other files. Thus, we identify a file by the rank with respect to replication, i.e., the file 1 is the most often replicated file and the file *M* is the least often replicated file. This allows the enumeration $\mathbb{F} = \{1, 2, ..., M\}$.

To close the loop between the active and passive behavior of a peer, we define a matching between queries $n \in \mathbb{Q}$ and files $m \in \mathbb{F}$. As a generalization of the *selection power* introduced in [16], we define a workload function that defines the matching probability of a given query to a given file:

Selection function $w_{select} : (\mathbb{F}, \mathbb{Q}) \rightarrow [0,1]$. The function $w_{select}(m,n)$ denotes the probability that file *m* matches to query *n*.

To derive the selection function w_{select}, we split the measured data into separate trace files each containing one hour of P2P traffic. We argue that a period of one hour on the one hand is long enough to gain sufficient confidence in the measures derived from the sub-traces. On the other hand, one hour is short enough to obtain a relative stable snapshot of the P2P network, which suffers only minor from system dynamics as peer arrival and departure. Nevertheless, we conducted similar experiments with sub-traces of two hours length and observe similar results.

We obtain *K*=1270 sub-traces. For each sub-trace, we calculate the set of recorded queries $\mathbb{Q}_k$ and the set of recorded files $\mathbb{F}_k$, $1 \le k \le K$. Again, we identify queries and files in $\mathbb{Q}_k$ and $\mathbb{F}_k$ by their ranks with respect to sub-trace *k*. For ease of exposition, let $q_k(n)$ be the query with rank *n* in sub-trace *k*, and $f_k(m)$ be the file with rank *m* in sub-trace *k*. Note that in general it does not hold $q_k(n) = q_l(n)$ and $f_k(m) = f_l(m)$ for $k \ne l$. We determine the number of unique queries *N* and unique files *M* as defined in Section 4.1 by calculating $N = \min_k(|\mathbb{Q}_k|)$ and $M = \min_k(|\mathbb{F}_k|)$ over all sub-traces *k*. Thus, we will characterize only the *N* most popular queries and *M* mostly replicated files in each set.

Given these definitions, we can approximate the selection function w_{select} by the relative frequency of the event that the file with rank *m* matches to the query with rank *n* in all sub-traces *k*, $1 \le k \le K$. Formally, we define an indicator function $f_m^k : \{1,...,N\} \rightarrow \{0,1\}$ for each sub-trace *k* by $f_m^k(n) = 1 \Leftrightarrow f_k(m) \in \mathbb{F}_{q_k(n)}$ for $(q_k(n), \mathbb{F}_{q_k(n)}) \in \mathbb{I}$, where the set $\mathbb{I}$ is given by the measurement study presented in Section 3. Using these definitions, the selection function w_{select} can be approximated by:

$$w_{select}(m,n) \approx \frac{1}{K} \sum_{k=1}^{K} f_m^k(n) \qquad (1)$$

Note that the selection function is not a probability mass function, as in general it holds $\sum_{m=1}^{M} w_{select}(n,m) \ne 1$ and $\sum_{n=1}^{N} w_{select}(n,m) \ne 1$ for given n and m, respectively. In fact, $\sum_{m=1}^{M} w_{select}(n,m)$ is the expected number of files matching query *n*, whereas $\sum_{n=1}^{N} w_{select}(n,m)$ is the expected number of queries matching file *m*. Both quantities can be obtained from the measured data to validate the selection function.

Figure 4 shows the selection functions for the files with ranks 1, 10, 100, and 1000 over the query rank. Obviously, the sum of the selection function values decrease with increasing file ranks. We argue that this decay is caused by the measurement setup. Recall that as illustrated in Figure 1, the measurement peer is not involved in all response message transmissions and thus cannot trace all matches between a file and a query. Since the probability that the measurement peer is involved in the transmission of response messages for highly replicated files is higher than for rarely replicated files, the sum of the selection function values decrease for increasing file rank. Consequently, we assume the sum of the selection function values of the most replicated file as appropriate for all files. We provide an experimental verification of the correctness of this assumption in Section 5. Because the selection function w_{select} depends only on the query rank n but is independent of file rank m, for ease of exposition we denote the selection function as $w_{select}(n)$.

Figure 4 shows that the shape of the selection function is similar for all file ranks. We find that it can be well modeled by a mixture distribution given by the summation of an exponential distribution and a normal distribution. Figure 5 shows that this mixture distribution fits well to the measured selection function. Thus, we model the measured selection function by:

$$w_{select}(n) = z\left(\lambda \cdot e^{-\lambda \cdot n} + \frac{1}{\sqrt{2\pi}\varpi} e^{-\frac{(n-\theta)^2}{2\varpi^2}}\right) \tag{2}$$

The parameter z=11.6 is obtained from the measurement study.

To employ the selection function for generation of a synthetic workload, we have to define a function, which preserves the shape and values of the selection function for varying number of unique queries N. To accomplish this task, we normalize the x-axes to 1 and re-define the selection function to characterize the matching probability based on the *relative rank* of the query with respect to all queries. Recall that the selection function is independent of the file rank.

Normalized selection function $w_{select,N} : \mathbb{Q} \rightarrow [0,1]$. The function $w_{select,N}(n)$ denotes the probability that an arbitrary file matches to the query with rank n in a system with N unique queries. Here, N denotes the input parameter of the workload model as defined before. The normalized selection function is given by:

$$w_{select,N}(n) = \tilde{z}\left(\tilde{\lambda} \cdot e^{-\tilde{\lambda} \cdot \frac{n}{N}} + \frac{1}{\sqrt{2\pi}\tilde{\varpi}} e^{-\frac{\left(\frac{n}{N}-\tilde{\theta}\right)^2}{2\tilde{\varpi}^2}}\right) \tag{3}$$

Here, the parameter set is derived by normalizing the fitted parameters of the measured selection function $w_{select}(n)$ with respect to the measured number of queries $N_{measured}$. That is:

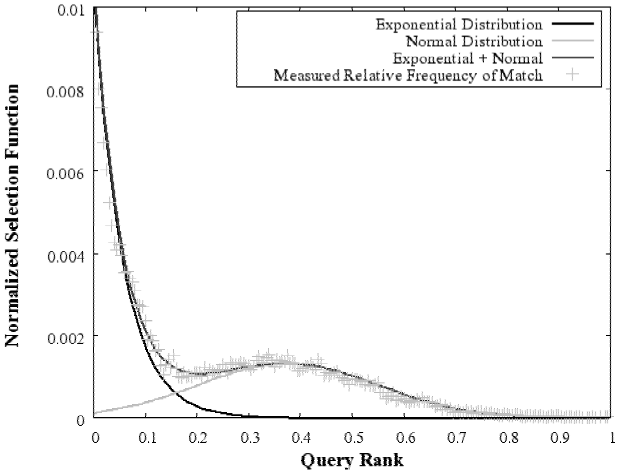

Figure 5. Measured normalized selection function and fitted exponential, normal, and exponential/normal mixture distribution for file rank 1

$$\tilde{z} = \frac{z}{N_{measured}};\ \tilde{\lambda} = \lambda \cdot N_{measured};\ \tilde{\varpi} = \frac{\varpi}{N_{measured}};\ \tilde{\theta} = \frac{\theta}{N_{measured}} \tag{4}$$

The values of the parameters $\tilde{z}, \tilde{\lambda}, \tilde{\varpi}, \tilde{\theta}$ are given in Table 2.

5 Comparative Evaluation of the Workload Model

To illustrate that the selection function presented in Section 4 provides an accurate model for the relationship between query popularity and file replication in a P2P file sharing system, we employ our approach in a simulation study of the Gnutella search algorithm. In this study, queries are flooded in a Gnutella- style overlay network. Each peer that stores a file matching to a query generates a response message. The message is returned on the reverse path of the query message.

We consider an overlay network consisting of P peers. Each single peer p maintains $CON_{MIN} \le c_p \le CON_{MAX}$ connections to other randomly chosen peers. In overlay construction, we assure that the resulting graph consists of a single connected component, i.e., there is an overlay path between any pair of peers. M files are assigned to the peers according to Section 3.3. Furthermore, each peer generates queries chosen from a set of N queries according to Section 3.2.

To obtain performance results, we connect a single measurement peer to $CON_{measure}$ randomly chosen peers. We record the workload characteristics considered in Section 3 from

Parameter	Value
$\tilde{z}$	0.00058
$\tilde{\lambda}$	18.1
$\tilde{\varpi}$	0.175
$\tilde{\theta}$	0.375
$N_{measured}$	20,000

Table 2. Parameters of the normalized selection function

Parameter	Value
Workload parameters	
P	8,636
M	11,078
N	11,545
T	3600 s
Network parameters	
CON_{MIN}	5
CON_{MAX}	10
$CON_{measure}$	20
TTL_{MAX}	7

Table 3. Parameters for the simulation study

the point-of-view of the measurement peer. For simplicity, we do not accurately model the routing of query and response messages, but assume that the measurement peer receives a query message if the distance between the measurement peer and the originator of the query is at most TTL_{MAX} hops. Similar, we assume that the measurement peer receives a response message with some probability if it is located on the shortest path between the originator of the query and the responding peer. Assume that $s+t$ shortest paths exist between these peers, s of them traversing the measurement peer and t traversing it not. Then, the measurement peer receives the response message with probability $s/(s+t)$.

To achieve sufficient confidence levels, we use the results of 40 sub-traces of one hour length and calculate average values for all performance measures. The parameters *P*, *N*, and *M* are roughly equal for all chosen sub-traces. *P*, *N*, and *M* for the simulation study are obtained by calculating the average over all sub-traces, which are shown in Table 3. We compare the average results from the traces with the average results of 40 independent finite time horizon simulations. 99% confidence intervals computed by independent replicates are included in all curves.

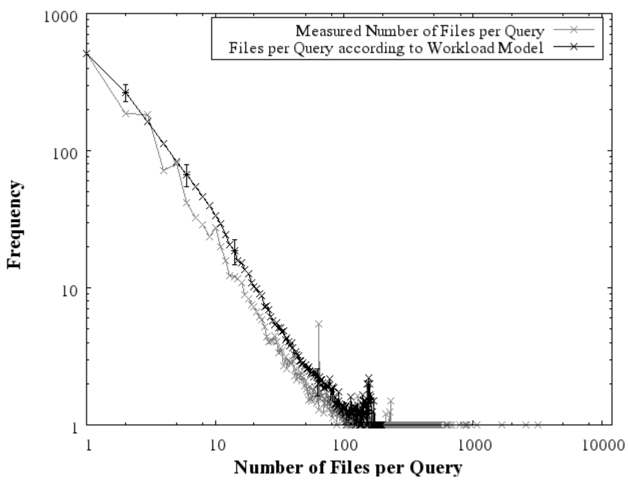

Figure 6. Comparison of file per query measured from trace and derived from synthetic workload model

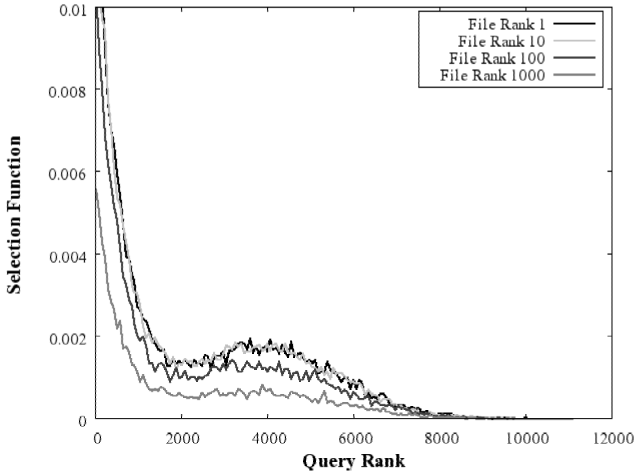

Figure 7. Selection function for various file ranks obtained from synthetic workload model

To validate the selection function presented in Section 4, we compare the distribution of unique files in responses to a query derived from the trace and the simulation, respectively. We use the absolute number of unique files obtained from the 40 sub traces for comparison. Recall that this measure is influenced by the network topology. Figure 6 plots the distribution of files per query for trace and simulation. The figure illustrates that the matching between files and queries is well modeled by this approach.

In an additional experiment, we analyze the matching between queries and files as defined by the selection function. Recall that the model presented in Section 4 assumes that the probability that a file matches a query is independent of the rank of the file with respect to file replication. This is justified by the fact that the measurement peer receives only few responses for rarely replicated files. Thus, the decay of matching probability with file rank is induced by an inaccuracy due to the passive measurement setup. In this section, we provide an experimental proof that this justification is valid.

We measure the matching probability as a function of file and query rank in 12,000 independent simulation runs. Figure 7 plots the matching probability as a function of query rank for different file ranks. We find that the matching probability decays with file rank similar to Figure 4. Recall that the matching probability is assumed independent of file rank when generating the matching according to Section 4. Thus, the decay in measured matching probability is induced by the number of copies of each file together with the overlay network topology, which prevents the measurement peer from receiving every response message. Figure 7 provides strong evidence that the matching probability is well modeled by the selection function presented in Section 4.

Conclusions

In this paper, we used the results of a comprehensive passive measurement study of the Gnutella peer-to-peer (P2P) file-sharing system to characterize user behavior in such system.

Specifically, we investigate which queries a user issues and which files a user shares. These aspects of user behavior can be characterized by query popularity and file replication, respectively. Using the results from the measurement study, we show that query popularity can be modeled by a Zipf-like distribution. Furthermore, file replication can be modeled by a Zipf-like distribution, too. As major contribution, we relate query popularity and file replication by characterizing the matching probability between queries and files. We showed that this matching probability is given by a probability function composed by a summation of an exponential distribution and a normal distribution. The probability function depends on the popularity rank of the query, but is independent of the replication rank of the file. Thus, the matching probability between queries and files is easily computable. To validate the presented approach, we conducted a simulation study of a Gnutella-style P2P file sharing system. The study illustrates that the load experienced by a peer in the simulated network closely matches the actual load recorded in the traces of the Gnutella network.

Both query popularity and file replication constitute important aspects of a detailed workload model that characterizes the behavior of a single user in a P2P file sharing system. In future work, we will use the results of our measurement to characterize other aspects of peer behavior, e.g., session length or think time between queries. Combined with the results of this paper, this will yield a comprehensive workload model that is appropriate for detailed simulation studies or even field studies based on software prototypes, supporting the design of novel approaches to large-scale P2P systems.

References

[1] E. Adar, and B. Hubermann, Free Riding on Gnutella, *Technical Report*, Xerox PARC, 2000.

[2] J.M. Almeida, J. Krueger, D.L. Eager, and M.K. Vernon, Analysis of Educational Media Server Workloads, *Proc. 11th Int. Workshop an Network and Operating System Support for Digital Audio and Video*, Port Jefferson, NY, 2001.

[3] J. Chu, K. Labonte, and B. Levine, Availability and Locality Measurements of Peer-to-Peer File Systems, *Proc. of SPIE ITCom: Scalability and Traffic Control in IP Networks,* 2002.

[4] E. Cohen and S. Schenker, Replication Strategies in Unstructured Peer-to-Peer Networks, *Proc. ACM SIGCOMM Conf. 2002*, Pittsburgh, PA, 177-190, 2002.

[5] Collab. Net, Inc. and Sun Microsystems, Inc., *jxta.org*, 2003. http://www.jxta.org.

[6] Gnutella Developer Forum, *Gnutella – A Protocol for a Revolution*, 2003. http://rfc-gnutella.sourceforge.net

[7] Z. Ge, D. Figueiredo, S. Jaiswal, J. Kurose, and D. Towsley, Modeling Peer-to-Peer File Sharing Systems, *Proc. IEEE Conference on Computer Communications (INFOCOM '03)*, San Francisco, CA, 2003.

[8] Mutella Hompage. http://mutella.sourceforge.net/

[9] Napster Homepage. http://www.napster.com

[10] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, A Scalable Content-Addressable Network, *Proc. ACM Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM '01)*, San Diego, CA., 149-160, 2001.

[11] S. Saroiu, K. Gummadi, and S. Gribble, A Measurement Study of Peer-to-Peer File Sharing Systems, *Proc. Conf. on Multimedia Computing and Networking 2002 (MMCN '02)*, 2002.

[12] Sharman Networks Ltd., *KaZaA Media Desktop*, 2003. http://www.kazaa.org.

[13] Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications, *Proc. ACM Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM '01)*, San Diego, CA., 149-160, 2001.

[14] K. Sripanidkulchai, The Popularity of Gnutella Queries and its Implications on Scalability, Featured on O'Reilly's www.openp2p.com website, February 2001.

[15] StreamCast Networks, *Morpheus 3.4*, 2003. http://www.morpheus.com.

[16] B. Yang and H. Garcia-Molina, Comparing Hybrid Peer-to-Peer Systems. *Proc. 27th Int. Conf. on Very Large Data Bases (VLDB '01)*, Rome, Italy, 2001.