

Characterizing Gnutella Network Properties for Peer-to-Peer Network Simulation *

Selim Ciraci, Ibrahim Korpeoglu, and Özgür Ulusoy

Department of Computer Engineering
Bilkent University
TR-06800 Ankara, Turkey

Abstract. A P2P network that is overlayed over Internet can consist of thousands, or even millions of nodes. To analyze the performance of a P2P network, or an algorithm or protocol designed for a P2P network, simulation studies have to be performed quite often, and simulation studies require the use of appropriate models for various components and parameters of a P2P network simulated. Therefore it is important to have models and statistical information about various parameters and properties of a P2P network. This paper tries to model and obtain the characteristics of some of the important parameters of one widely used P2P network, Gnutella. The methodology to derive the characteristics is based on collecting P2P protocol traces from the Gnutella network that is currently running over the Internet, and analyzing the collected traces. The results we present in this paper will be an important ingredient for studies that are based on simulation of P2P networks, especially unstructured P2P networks.

1 Introduction

Peer-to-peer (P2P) systems enable formation of huge overlay networks over Internet and allow users to become active participants in these networks. Each node is called a servent in a P2P network and acts both as a server and a client. There are several types of P2P networks, including unstructured P2P networks [2], loosely structured P2P networks, and structured P2P networks [1]. Unstructured P2P networks can further be divided into three types, which are pure, hybrid, and centralized. In pure unstructured P2P networks, each node has equal responsibilities. In other type of unstructured P2P networks [3], on the other hand, some nodes can take special responsibilities like holding an index of the resources shared by the neighboring nodes.

Unstructured P2P systems are good candidates for serving large number of Internet users due to their distributed nature. The major problem with unstructured P2P systems, however, is efficiently locating the requested resources (efficient search). The current mechanism for searching is based on flooding the

* This work is supported in part by The Scientific and Technical Research Council of Turkey (TUBITAK) with Grants EEEAG-103E014 and 104E028.

query messages and therefore it is not efficient. There exists a substantial amount of research on improving the performance of unstructured peer-to-peer networks, including the performance of search operations, and there are many methods proposed. Evaluating the methods and their performance, however, is not easy. The number of nodes constituting a P2P network is huge and there are lots of parameters that should be considered, which make analytical approaches quite difficult to use in the evaluations. Therefore we have to resort to simulation models quite often. But building accurate and correct simulation models requires accurate modeling of the properties and workloads of real-life systems that are simulated. Therefore, it is important to characterize and model the parameters and workloads of real P2P systems that are operational, in order to be able to simulate them accurately.

In this paper, we aim to characterize some of the important parameters of an operational unstructured P2P network, the Gnutella network, by examining the protocol traffic traces that we have collected from the Gnutella network. In analyzing and summarizing these traces, we have focused on the characterization of keywords (their numbers and types) in queries, time-to-live (TTL) values in query messages, peers' contribution to the network, and the characteristics of repeated queries.

The paper is organized as follows. In section 2, some of the related work is described. Then, in section 3, our Gnutella crawler that is used to collect traces from the Gnutella network is described together with our methodology in collecting the traces. In section 4, the results derived from the traces are presented, and finally in section 5 our findings are summarized.

2 Related Work

There exist several studies on the measurement and analysis of several P2P networks. The study on [4] lists some of the important parameters that should be considered when simulating a P2P file sharing network. In this study, models for some of the parameters are derived from real world observations, and the parameters considered are separated into two groups. The first group of parameters are related with the distribution of resources in the P2P network, and the second group of parameters are related with modeling the behavior of peers. The main difference of our study from [4] is that we try to characterize P2P network parameters using traces collected by custom P2P crawlers. We also investigate some parameters that are not investigated in [4].

The authors of [5] has conducted an analysis of the Gnutella network using crawlers, like we did. They logged for an hour the Query and Query hit messages seen at three different points on the Gnutella network. The study of the logged messages is focused on the detailed analysis of repeated queries, the TTL values seen in the queries, and the inter-arrival times of submitted queries. In this paper, we also analyze some aspects of repeated queries and the TTL values of user submitted queries. But we are more focusing on the characterization of initial TTL values set in the queries, and on the characterization of inter-arrival

times of repeated queries. A similar study to [5] is presented in [6]. That study, however, is more focused on content analysis of queries. It derives and lists some popular keywords that are used in submitted queries. In this aspect, the work also resembles to what we did, but we are also trying to find a model for the repetition count of popular keywords.

The study presented in [7] also uses crawlers to collect message traces from Napster and Gnutella networks. It plots cumulative distributions of peer characteristics such as the number of resources shared, the uptime of peers, and the bandwidth capacity of peers. In this paper, we also focus on similar parameters such as the number of resources shared by peers, but we also try to come up with a model that can be used to generate similar values for these parameters in simulation studies.

3 Methodology

To derive information about various parameters of a Gnutella network, we followed a methodology similar to the one described in [7]. We programmed a custom Gnutella crawler to collect Gnutella network traces. Using the crawler we gathered large sets of data and logged them on a local disk. The logged data includes various Gnutella protocol messages that suit our measurement goals. By using the addresses obtained from the logged messages, we also probed numerous nodes to get an idea about their states and uptimes.

In this section, we first briefly introduce the Gnutella architecture and its protocol messages. Then we describe briefly our Gnutella crawler that is used to collect Gnutella protocol messages transported over a portion of the Gnutella network. We then introduce and describe some of the P2P network parameters which we are trying to characterize and estimate using the message logs we obtained via our crawler.

3.1 Gnutella

In Gnutella network, peers form an overlay network over Internet by opening point-to-point TCP connections to each other. To join the overlay, a newcoming peer has to discover a small subset of the active overlay participants. This discovery is done by querying the hostcaches, which hold the IP addresses of some of the high-uptime participants. Each Gnutella compatible P2P client comes with a set of predefined hostcache addresses. After discovering a set of the peers to join, a newcomer initiates Gnutella handshake with a peer in that set (Figure 1). During this handshake, both the newcomer peer and the peer that is already part of the Gnutella network indicate to each other the Gnutella protocol version they are using and the extensions they support [2]. If the peer that is already part of the Gnutella network can accept the connection request from the newcoming peer, it indicates this by sending an OK message. If, on the other hand, the peer cannot accept the connection, it indicates the reason why it cannot accept the connection and provides the newcomer with a set of peers it knows.

This way the newcomer can discover other peers without further querying the hostcache.

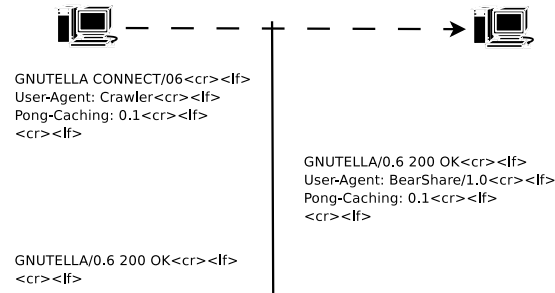


Fig. 1. A successful Gnutella handshake between two peers.

After a successful connection establishment, peers start exchanging Gnutella protocol messages. A Gnutella message header consists of a *global unique identifier (GUID)* field, a *time-to-live (TTL)* field, a *hops* field, a *payload type* field, and a *payload length* field. The GUID is used to overcome routing loops that may occur in the overlay. To prevent routing loops, a peer receiving two messages with the same GUID ignores the second one. Each peer receiving a Gnutella message increases the hops count value in the message by one and also decreases the TTL value by one. When the TTL value of a message reaches to zero, the message is not forwarded anymore. The payload type field is used by peers to distinguish different types of Gnutella messages. There are five types of Gnutella messages which are *Query*, *QueryHit*, *Bye*, *Ping*, and *Pong* messages.

A Query message contains the user submitted query string as its payload. A peer receiving a Query message checks its shared resources for a match to the query string included in the Query message. If the peer has resources that match the query string, it sends a Query Hit message back. The Query Hit message is set the same TTL value as the *hops* field of the corresponding Query message. The payload of the Query Hit message contains the physical address of the originator and the names of the resources that match the corresponding query.

The Ping and Pong messages are used to exchange topological information. When a peer receives a Ping message, it answers back with at least 10 Pong messages each containing the physical addresses of other peers that are collected again by sending Ping messages. Bye is used by a peer to indicate its disconnection from the network to its neighbors.

3.2 Gnutella Crawler

Our Gnutella crawler is written in Java and follows the Gnutella protocol specification version v0.6 [2]. First, the crawler connects to the HTTP address *gweb-cache2.limewire.com:9000/gwc* to collect physical addresses of some active peers. It then starts opening connections to those peers and also builds its own host-cache from the physical addresses collected via unsuccessful connection attempts and Pong messages. After connecting to three peers successfully, the crawler starts monitoring and logging Gnutella messages considering the parameters we are going to discuss.

3.3 Measured Parameters

The simulation of a Gnutella network requires consideration of a lot of parameters. We focused only on a subset of all possible parameters and tried to understand the nature of the values of these parameters in the Gnutella network. We now introduce the parameters we focused on, and describe how the related traces are collected to obtain the characteristics of these parameters.

Number of keywords contained in a query: For semantic routing techniques, keywords in a query define routing rules for that query. Thus, the more keywords a query has, the more information the routing technique can extract about the query's route. It is widely believed that P2P users submit short queries consisting of one or two keywords, so it's difficult to apply semantic routing techniques. To test this belief, we have programmed the crawler to collect 10 thousand queries from five different connection sets (each set consisting of different nodes). After collecting the data, the queries are tokenized with "._*()";:;!?" delimiters to extract the keywords and then each keyword is counted. To combine the counts from different connection sets, the averages of the counts is taken.

Repetition rate of keywords in queries: It is a fact that in P2P networks there exist some popular resources which are queried a lot. Many protocols that try to improve search quality rely on repetition rate of keywords in queries. Therefore it is important to develop a model for popular keywords for such techniques.

To develop this model, we have used the tokenized queries of the previous parameter and hashed each keyword using Java's string class, which hashes strings by adding the integer values of each character in a string. These hashed keywords are used as a key to index the hash table holding the number of accesses made to the cells. We have given the highest rank of 1 to the mostly accessed cell, which in turn is the keyword with the highest repetition.

Initial TTL values of queries: For P2P simulations, the initial TTL values set in Query messages play an important role, since Query messages can travel longer distances with a higher TTL value, which increases the chance of finding the resources requested by the query. Gnutella protocol specification [2] states that TTL values in queries should be set to 7 and many P2P simulators follow this specification. However, the fact that many Gnutella clients today use shorter

initial TTL values makes TTL an important parameter to consider to achieve realistic P2P simulations.

To keep track of TTL values, while collecting query data for the previous parameters we have also programmed the crawler to log the *TTL* and *hops* values of the received queries. The initial TTL values are calculated by adding these two values. Again averages of several collected data sets are used to obtain the final estimates.

Peers' contribution to the network: Distribution of resources to peers in a P2P simulation should also be handled carefully, since the query hit rate is directly affected by this parameter. Some previous studies show that %25 of the Gnutella peers do not share any files at all, and %7 of peers share 100 files [7].

To collect the required data to estimate the distribution characteristics of resources, the crawler has been programmed to collect 10 thousand Pong messages from five different connections sets. The collected Pong messages contain the physical addresses of the nodes sending the Pong messages, and the total number and size of resources shared by these nodes.

Query Hit to Query ratio: Although peers' contribution to the network greatly affects the Query Hit messages returned to Query messages, the popularity of the shared resources is another important factor that can affect the Query Hits, since popular resources will be queried more than the other ones. Therefore it is also important what kinds of files are shared by a peer. It is, however, hard to model the popularity of shared resources, but collecting the number of Query messages with matching Query Hit messages in the Gnutella network may help in characterizing the popularity of resources. For example, if we find that $x\%$ of the Query messages in the collected data set have a matching Query Hit message, we can then adjust the popularity parameter in a simulation so that the chance of getting a Query Hit to a Query message in the simulation is $x\%$. It is easy to find and count the Query Hit messages corresponding to Query messages submitted in the Gnutella network. We should just match the GUID fields stored in the Query and Query Hit messages.

Repeated queries: When the P2P network does not return any results to a query submitted by a user, the query is re-submitted by the user or the P2P client software. Thus, it may be important to model this behaviour for simulation of caching systems.

In order to find out how many queries are repeated in a five different query sets each containing 10 thousand queries, we have hashed the query string in a Query message together with the *hops* value of the message, again by using Java's string class. If two different queries are hashed to the same cell, then that query is marked as a repeated query. Although it is impossible to know which peer has submitted the query when the hops value is greater than 1, two queries with the same query string and the same hops value have a very high probability of being repeated, thus we have used this method to recognize repeated queries.

TTL values of repeated queries: When a user of a P2P system re-submits a query, it provides some advantage for the P2P client to send the query to the network with a larger TTL value. Although Gnutella specification does not

mention this, some clients may have adapted this approach in order to increase search quality. This makes it important to analyze the TTL values in repeated queries. In order to analyze this behaviour, the crawler also logs the TTL values in queries that are recognized as repeated queries.

4 Results

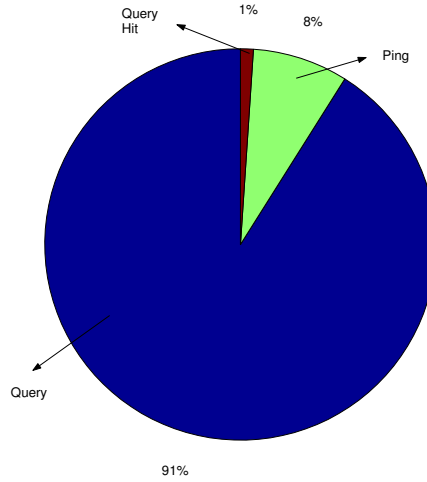


Fig. 2. The Gnutella protocol messages observed in trace data and their fraction of occurrences.

In this section we present our results about the characteristics of the parameters that we have described. Before that, however, we would like to present a pie-chart describing the ratio of Gnutella protocol messages seen on the traces (Figure 2). In this chart, we did not include the Pong messages since they are sent whenever a node receives a Ping message. The overhead of flooding is clearly seen in the figure; %91 of the Gnutella traffic consists of Query messages. Thus the need for a protocol that reduces this overhead is clear.

In Figure 3, the distribution of the number of keywords submitted in a query is shown. Our analysis of the related traces shows that 68477 queries out of 100 thousand queries contain less than 5 keywords. We found 4 as the mean number of keywords that can be seen in a query. Queries with just one keyword constitute the 10% of all queries we analyzed. The Figure 3 indicates that users tend to submit more descriptive queries instead of submitting single-keywords queries. It is also interesting to notice that 1561 queries out of 100 thousand queries contain more than 7 keywords which makes around 1.5% of all the queries analyzed.

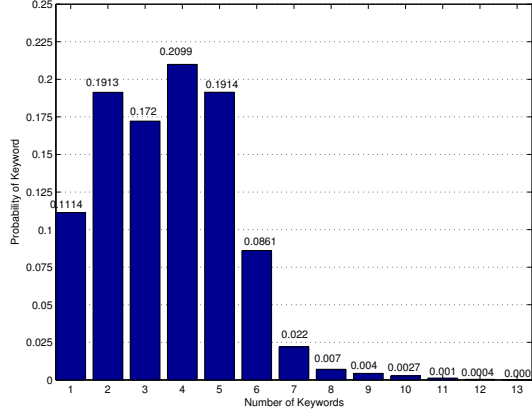


Fig. 3. Distribution of number of keywords seen in query messages.

Figure 4 shows the repetition count of keywords in user submitted queries. In plotting the graphs in the figure, we first ranked all the keywords with respect to their repetition count. In Figure 4-a, the x -axis is the rank of the keywords, and the y -axis is the repetition count of the keywords with respect to those ranks. The keyword that has the highest repetition count has rank 1. The analysis of this plot shows that the repetition count of keywords obeys a power-law distribution with respect to the rank of keywords. We think this is due to popularity of some keywords. For example, in our traces, keywords “mp3” and “divx” are the two most popular keywords and they have very high repetition counts (i.e. we can see them in many queries). Since the curve on the graph is steeply decreasing, we only plotted the repetition counts up to rank 1000, otherwise it was difficult to identify the curve on the graph. To better show that the repetition count of keywords obey a power-law distribution, we plotted the repetition count versus rank of keywords in logarithmic scale, and fit a polynomial with degree 1 to the curve obtained in this manner. The Figure 4-b shows the plot in logarithmic scale with the fitted polynomial (in this plot we did not limit the rank). The fitted polynomial has coefficients -1.028 and 4.74 (i.e. it is the line described by equation $y = -1.028 \times x + 4.74$).

The distribution of initial TTL values observed in Query messages are shown in Figure 5 as a pie-chart. As can be seen from the figure, majority of the Gnutella clients (89%) set the initial TTL value to 4 in Query messages. The clients setting the initial TTL value to 3 constitute around 11% of the peers. The number of clients setting the initial TTL value to something else is less than 1% and therefore negligible. We also tested what happens if a client tries to submit Query messages with larger initial TTL values than 4. For this we modified our Gnutella client so that it submits queries to the network with TTL values larger than 4. We have noticed that majority of the clients around us have lowered the TTL value to 4. We believe that Gnutella developers have taken such an

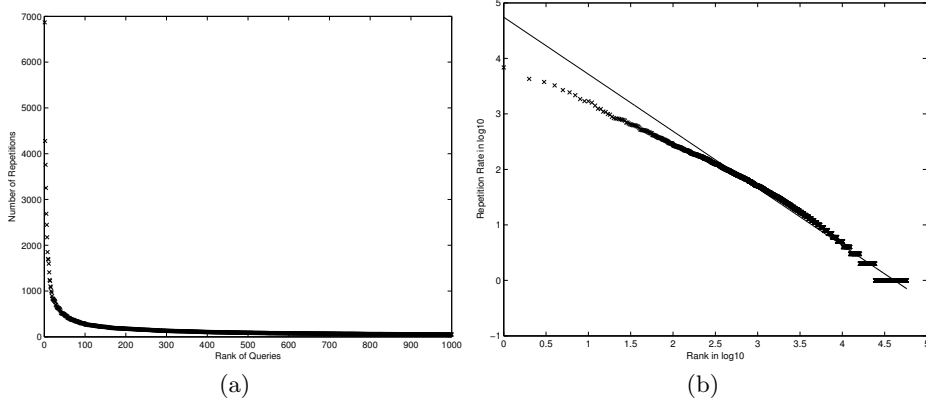


Fig. 4. Repetition count of keywords. a) Repetition count of keywords versus the rank of keywords. Keywords are ranked according to their frequency of occurrence in query messages. b) Log of repetition count of keywords versus log of rank of keywords.

action to lower the overhead introduced by the flooding mechanism used for disseminating the queries.

In Figure 6, we show the cumulative distribution function of number of files shared by a peer. On the x -axis we have the number of files shared, and on the y -axis we have the fraction of peers sharing number of files that is less than or equal to the corresponding value indicated on the x -axis. From the figure we see that 50 peers out of 420 peers share zero files. In other words, nearly 10% percent of peers do not share any files. The figure also reveals that only around 5% of peers share more than one thousand files. These are not surprising results since it is a quite well-known fact that only a small percentage of peers in a P2P network share huge numbers of files. It is also interesting to notice that although many peers indicate that they share small number of files, these shared files are quite large in size (around 2 GB). This leads us to believe that in Gnutella network users tend to search and download large files which in turn causes peers to share large files.

Although Query to Query Hit ratio greatly depends on the queries submitted, from Figure 2 one can see that Query Hit messages constitute only 1% of the overall P2P message traffic observed in the traces. This is quite a small fraction, but this is not very surprising since a Query Hit message is sent in a unicast manner, whereas the corresponding Query message is flooded.

We also looked how many times a query string is repeated by a peer in while submitting queries. A query string can be repeated by a peer because the results obtained in previous query submissions may not be found satisfactory by the peer. Out of the 100 thousand queries observed, we have identified 15678 queries as repeated queries. This constitutes 15% of all the queries observed. Figure 7 shows that majority of the queries are repeated twice (81% of all queries). We

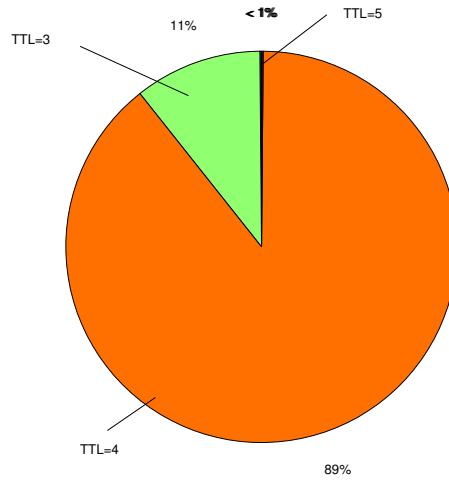


Fig. 5. Initial TTL values seen in query messages and their percentages. Most queries observed in the traces have an initial value of 4.

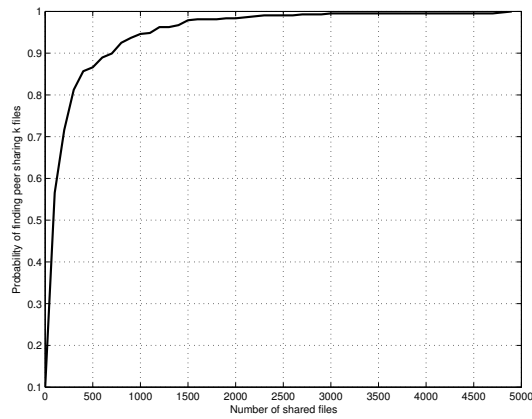


Fig. 6. Cumulative distribution function (CDF) of the number of files shared by a peer. Most of the peers (95%) share less than 1000 files.

have found that only 2 queries are submitted to the network more than 5 times. These two queries have all "?" as query strings, which we believe are used by peers to discover all the names of the resources shared by their neighbors, although nothing about this is mentioned in Gnutella protocol specification. Our inter-arrival time analysis for repeated queries shows that on average there is a 1242291.09 msec time interval between the repeated queries. This corresponds to around 21 minutes between each repeated query, which is a reasonable time, since a user re-submits a query after the arrival and inspection of the previous results. Our TTL analysis for repeated queries shows that the initial TTL values of these 15678 repeated queries are not increased by the clients submitting these queries. Given that majority of the queries are repeated only twice, we can say that a Gnutella user is satisfied with the results after a second submission that comes after a sufficiently large inter-arrival time. Since a P2P network is quite dynamic, during this time the topology of the P2P network might have changed and it might be now possible to reach to some more resources in the second query submission. Therefore, there is no need to increase the initial TTL in the re-submitted queries for the purpose of enlarging the search horizon.

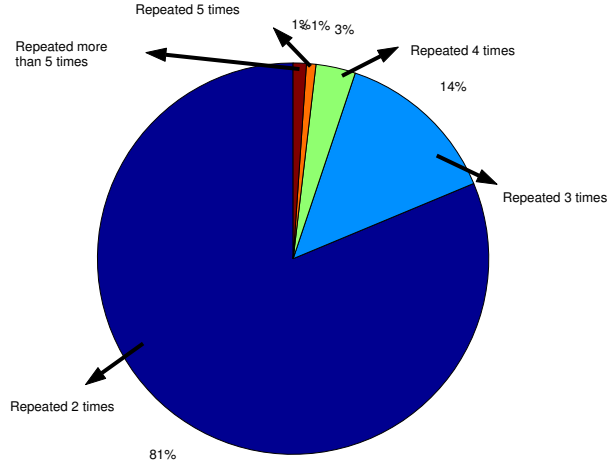


Fig. 7. The count of repeating the same query string by a peer. 81% of peers that repeated a query sting have repeated the string 2 times.

5 Conclusion

In this paper we derived characteristics of some important Gnutella network parameters based on real network traces obtained from the current live Gnutella network. As already mentioned by several studies, we have verified that a large

portion of Gnutella protocol messages seen on a Gnutella network is constituted by Query messages which are disseminated through a simple and inefficient flooding mechanism. This clearly indicates the need for more clever algorithms for disseminating queries in unstructured P2P networks to reduce the messaging overhead and to provide better scalability.

Our results also indicate that most submitted queries contain query strings that consist of multiple keywords, as opposed to the common assumption in various simulations that a query consists of a single keyword. We also found that repetition count of keywords seen in a P2P network obeys a power-law distribution with respect to the rank of keywords where the keyword that is repeated the most has a rank of 1. We also verified the fact that not all peers contribute to a P2P network at the same level. A small portion of peers share a large portion of all files available in the network. Our traces also revealed the fact that the same query string is not repeated too much by the same peer, and the peer does not increase the initial TTL (time-to-live) value in a repeated query to enlarge the search horizon. We have found that most submitted queries have an initial TTL value of 4, and even though a peer submits a query with a larger TTL value, the neighboring peers immediately reduce the TTL value to 4.

We think that our findings can be important for P2P network simulation studies that are looking for models and information about some of the important parameters of P2P networks.

References

1. Stephanos, A., T.,: A Survey of Peer-to-peer File Sharing Systems. WHP-2002-03, Athens University of Business and Economics, 2002.
2. Gnutella protocol v0.6. Available at <http://rfc-gnutella.sourceforge.net/developer/testing/index.html>
3. Kazaa <http://www.kazaa.com>
4. Schlosser, M., T., Condie, T., E., and Kamvar S., D.,: Simulating a File-Sharing P2P Network. First Workshop on Semantics in P2P and Grid Computing, December, 2002
5. Markatos, E., P.: Tracing a large scale Peer-to-Peer System: An hour in the life of Gnutella. In 2nd IEEE/ACM Int. Symp. on Cluster Computing and the Grid, 2002.
6. Zeinalipour-Yazti, D., and Folias, T.: Quantitative Analysis of the Gnutella Network Traffic. TR-CS-89, Dept. of Computer Science, University of California, Riverside, June 2002
7. Saroiu, S., Gummadi, P., K., and Gribble S., D.: A Measurement Study of Peer-to-Peer File Sharing Systems. Proceedings of Multimedia Computing and Networking 2002 (MMCN'02), San Jose, CA, January 2002.
8. Patro, S., and Hu, Y., C.,: Transparent Query Caching in Peer-to-Peer Overlay Networks. In Proceedings of the 17th International Parallel and Distributed Processing Symposium (IPDPS), Nice, France, April 22-26, 2003. (119//407 29%)