Search Methods in P2P Networks: A Survey

German Sakaryan, Markus Wulff, and Herwig Unger

Computer Science Dept., University of Rostock,
18059 Rostock, Germany
{gs137,mwulff,hunger}@informatik.uni-rostock.de

Abstract. The peer-to-peer (P2P) file sharing applications have gained a tremendous popularity and now they have millions of users worldwide, since they were introduced in 1999. Due to chaotic structure, achieved scale and network dynamics, they mostly employ a flooding-based search to locate required files and therefore they are the main source of Internet traffic. Thus, the study and development of P2P systems is an important research challenge.
This article presents a survey of existing approaches to organize operations of P2P file sharing systems. It gives a classification of existing protocols and discusses the advantages and disadvantages of each technique. It shows that both network structure and search algorithm influence the operations of P2P applications.

1 Introduction

The peer-to-peer (P2P) file sharing systems are the large scale distributed applications devoted to file exchange among a large number of Internet users.

In contrast to popular client-server applications, end user machines in P2P systems are involved in network operations by providing their resources (e. g., storage space, bandwidth and computer power) and acting as both clients and servers. At the same time, they are not required to have permanent IP addresses as it is necessary for servers in client-server Internet applications. Therefore, they have significant autonomy, i. e., they can join and leave a network [1]. Utilizing user machines in P2P systems helps to distribute the costs of file sharing among the huge number of individual participants [2]. Any centralized server that provides both content and directory service is extremely expensive to run.

Built at the application level, P2P systems consist of the peers connected by a number of point-to-point connections (typically TCP). Due to the chaotic unstructured nature and transient peer population, P2P systems mostly employ flooding-based search mechanisms when many peers are probed whether they have a requested file or not. Such a flooding is the main source of the Internet traffic [3,4], which limits further system growth.

In this paper, it is intended to analyze existing approaches used to search files in P2P systems. More particularly, it is intended to discuss the influence of the methods on the search and network structure as well as methods applicability to real P2P applications. The presented methods are categorized depending on how search is realized and how the network structure is managed.

Accordingly, the article is organized as following. Section 2 presents a classification of the existing search methods in P2P systems. It discusses the advantages and limitation of these techniques. Section 3 presents a detailed description of P2P protocols. The paper ends with a conclusion.

2 A Classification

The P2P protocols can be classified depending on how the search is realized and how the network structure is organized and maintained (Fig. 1).

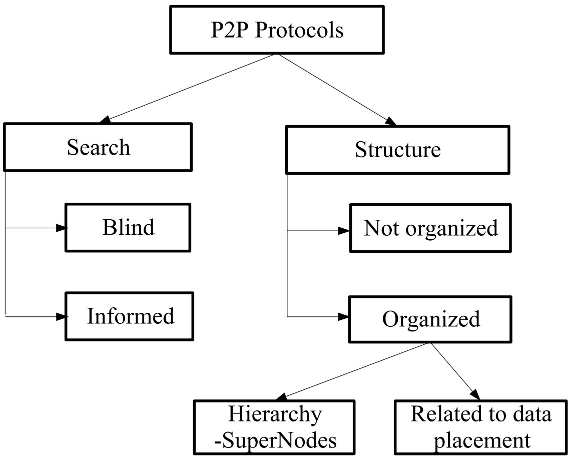

Fig. 1. A classification of P2P protocols

Existing search methods can be split into two main categories depending on how search query forwarding decisions are made:

- **Blind Search.**
  In a blind search, peers has no information related to file location. Therefore, other peers are randomly probed whether they have a requested file or not. Blind search protocols are simple and require peers to maintain a minimum amount of knowledge about network organization.
- **Informed Search.**
  In these systems, peers maintain additional information about file locations which can be useful for the search. For example, a peer may have an index of files offered by other peers connected to it. Based on this additional information, a peer decides which peers should be contacted.

A structure of P2P network is organized by point-to-point connections each peer has. Depending on the criteria used to form these connections between peers, the existing protocol are categorized as follows:

- **Structure is not organized.**
  Connections between peers are made mainly chaotically. They do not represent any dedicated network structure.
- **Structure is organized.**
  Protocols build a dedicated network structure which is an important part of the network operations. Therefore, the connections between peers are set in a special manner to achieve required network organization. Currently, there are two main kinds of protocols which build network structure. In the first one, all the peers are divided into powerful peers which are mainly responsible for network operations and weak peers connected to them. Accordingly, the resulting structure is a two-level hierarchy.
  The second one builds a network structure in accordance to data placement. In this way, connections between peers are related to the data shared by those peers (e. g., to connect peers which offer the similar content).

3 The Main P2P Protocols

The main P2P protocols can be divided into three main groups based on the classification presented above. The protocols of the first group employ blind search with/without network structure building. The majority of existing P2P file sharing applications use protocols of the first group. The second group of P2P protocols uses informed search without structure organization. The last group of protocols employs informed search techniques in addition to network structure organization.

The P2P applications and their protocols are presented in the following sections in groups.

3.1 Blind Search With/Without Structure Building

Gnutella [5] is decentralized both in the terms of file storage and file search. Therefore, it belongs to pure P2P systems. As far as the connections between peers are formed mainly chaotically, Gnutella belongs to unstructured systems in short of complete knowledge. Searching in such networks is mainly done by random search, in which various peers are randomly asked and probed if they have a target file. In Gnutella, a peer initiates a controlled flood of the network by sending a query message to all of the peers connected to it. When receiving a query, a peer checks if it has a requested file. If so, it forwards a special message back toward the search initiator [6]. No matter whether a file is found or not, the peer continues to flood the query. To limit generated traffic, each message can be forwarded only *time to live (TTL)* times. The advantage of such unstructured system is that they easily react to the high dynamics of P2P networks, peers can join and leave network without any notification. The obvious disadvantage is that it is difficult to find desired files without flooding queries which influence the scalability of such systems and significantly increase traffic.

FastTrack [7] protocol is a second generation of P2P protocols. It utilizes the heterogeneity between peers (computer power, bandwidth and availability) to form the two-level organization of a P2P network. FastTrack is based on the Gnutella protocol and extends it with the addition of *supernodes* to improve scalability.

It is used in KaZaa P2P application [8] where a small number of powerful peers (supernodes) handle traffic and routing on behalf of weak peers connected to them. KaZaa peers dynamically elect supernodes which form an unstructured network and use query flooding to locate content. Regular peers connect to one or more supernodes to initiate query and actually act as clients to supernodes [9].

Even though FastTrack utilizes heterogeneity between peers, it still relies on random search which is characterized by big volume of generated traffic. In addition, a limited number of peers responsible for traffic handling and message routing makes P2P networks more vulnerable to planned attacks.

Random Walks [10,11] is a range of blind search methods which benefit from using the properties of overlay topologies. To search for an object using a random walk, a peer chooses a neighbor randomly and sends the query to it. Each neighbor peer repeats this process until the data are found. In addition to random walks, the proactive replication algorithms [10] can be used. It was proposed to replicate files in proportion to the square-root of their query rate; this replication scheme can be implemented in a distributed fashion by replicating objects for a number of times proportional to the length of the search. The proactive replication may take place on the peer even though the peer has not requested the replicated data.

In the case of power-law random graphs [12], it was observed that high degree nodes experienced correspondingly high query loads. Even though the proposed approach helps to decrease generated traffic, it realizes the idea that, due to massive replication, data can be found in the nearness (limited number of hops) from the search initiator and if message will be broadcasted only to part of all the neighbors, the data still can be found almost of the same probability. The disadvantage of this approach is a random choice of a neighbor. The peers available for a long time could be chosen more often.

The **Logical Clustering** [13] approach tries to improve the information management in P2P networks by building dynamic cluster structures. The actual network topology is not changed. The logical clusters are built by circulating messages (wanderers). This provides the basis for a QoS (quality of service) for this high dynamic distributed system, i. e., each peer can be visited by a wanderer within a given time frame.

As mentioned above, the P2P networks can become very huge. The clustering supports a division of the network in such small groups (clusters) that a QoS regarding information management which fulfills certain time requirements can be provided. The time in which a peer wants to be supplied with information is the basis for the creation of the clusters Furthermore, the basic infrastructure for a cooperation between the peers is provided by applying the clustering techniques. Therefore, the clusters described here are defined as a subset of peers in a P2P network which can be covered by a wanderer. This wanderer must be able to visit all peers in that sub-network within a given time-frame.

The algorithm running locally on every peer is able to build the clusters by using just the knowledge about the network which is kept in the local warehouse. The communication between the disjoint clusters is organized by wanderers as well. For this purpose, every cluster has a dedicated peer. These peers are used to build higher-level clusters based on the same mechanisms. The clusters of these peers are connected by wanderers of the next higher level and so on. This results in a tree like structure, which can be used to spread messages (e. g., search requests) in the network.

The advantage is that the number of messages in the system is limited by using wanderers as tokens. The logical tree of clusters is used for search purposes.

3.2 Informed Search Without Network Structure Organization

Query Routing Protocol (QRP) [16] proposes to make informed search instead of network flooding. Peers are responsible for creating their own routing tables which include hashed keywords and regularly exchange them with their neighbors. These keywords are used to describe locally offered files. A Bloom filter [17] is usually used to code a set of keywords in a binary array which is suitable for transmission.

The computed routing tables are propagated up to several hops away from the peer it belongs to. If a peer receives routing tables from its neighbors, it can merge the tables and propagate the merged table further. In this scheme, the peer may decide to which of the neighbors it is worth routing a search query because merged routing tables of its neighbors are available. To decrease a number of hops required to find required data, routing table should be propagated as far as possible. At the same time, the limitations of bandwidth and increasing (with bigger number of propagating steps) ratio of false positives (mistakes with membership checking [17]) limit this idea. In addition, far propagations decrease actuality of information due to network dynamics.

Routing Indices (RI) [18] is an approach which employs ideas similar to QRP. In contrast to QRP, RI does not use Bloom filters to summarize content of neighbors. Instead, they use categorization of documents, which is a complicated task for real applications. Each peer maintains routing indices for each of the neighbors, indicating how many documents of which groups could be found through that neighbor. If a new connection is installed, a peer aggregates all routing indices of its neighbors, adds information about documents shared locally and sends an aggregated RI to a new neighbor. To keep data actuality, peers should inform neighbors of the changes in RI caused by addition or removal of documents or arrival/leaving of other neighbors. For practical purpose, this might cause frequent retransmission of data since network is constantly changing. Routing is done by the evaluation of "goodness" of neighbors against a query. If more documents could be found through particular neighbor, that neighbor is chosen.

Adaptive Probabilistic Search (APS) [19,20] uses the additional information about neighbors to make routing decisions. APS utilizes the feedback from previous searches to make future search more focused. More particularly, each peer keeps a local index describing which objects (files) were requested by each neighbor. The probability of choosing a neighbor to find a particular document depends on previous search results. The update takes a reverse path back to the peer search initiator and can take place after success or failure, adjusting probability accordingly. In this way, a network demonstrates a self-learning property which is supported by the discovered fact that many of the requested files are close to the requesters [4]. In contrast to RI or QRP, no additional overhead is required to obtain additional information about neighbors. Therefore, it could be better adapted to the changing topologies. The used approach leads to the situations when popular files could be located very fast, while other files could be hardly located. Other limitations are caused by the fact that first discovered peer (which has a requested object) might be more used for future routing and might experience more load. At the same time, other peers which are closer (from time or path length) could be ignored.

3.3 Informed Search with Network Structure Organization

Systems with Distributed Hash Tables (DHT). All the systems described above were unstructured so that the connections between peers are made chaotically, and data placement is completely unrelated to the structure formed by connections between peers.

To overcome the scalability limitations of flooding-based unstructured systems, the recently introduced systems like CAN [21], Chord [22], Pastry [23] and Tapestry [24] use another approach to the routing and topology organization of P2P networks. This approach employs the idea of *distributed hash tables (DHT)* functionality—mapping "keys" onto "values"—on Internet-like scale.

In DHT systems, files are associated with a key which is produced by hashing, for example, the file name or the file content. The range of the output values of the hash function forms an ID space. Every peer in the system is responsible for storing a certain range of keys (or partition of ID space). The structure is formed by routing tables locally stored on individual peers. A table includes a list of other peers with addresses and range of keys they are responsible for. Such systems are highly structured. The topology is tightly controlled and files (or information about files) are placed at the precisely specified locations defined by their keys. Depending on the application, the network structure and routing may be different (for details see [25]). Due to careful data placement and tightly controlled topology, DHTs have a scalable routing ($O(\log n)$ steps), where n is the size of a network (number of peers).

The important disadvantages of DHTs concern to the resilience of the system [26]. Since tightly maintained structure is important for correctness of operations, the structured design is likely to be less reliable in the face of a very transient user population.

Another important key problem of DHTs is load balancing. Due to significant file popularity skewness [9,27], peers responsible for storing pointers to the most popular files (or keywords) may become *overloaded*, since the majority of requests for the most popular content come to those peers.

In addition, retrieving content requires the knowledge of the exact file identifier, which is problematic in real P2P file sharing applications. All these require additional functionality which is missing in the original design of DHT systems.

A Content-oriented Approach to Network Structure Building and Search [28] does not require tightly controlled network structure and therefore copes with network dynamics better than DHTs. It includes two main concepts: content-oriented search and content-oriented structure building.

Content-Oriented Search

To avoid flooding, each peer stores content summaries of its neighbors. The summaries of peers can be represented as Bloom filters to code the set of keywords used in file names offered by those peers. If a network is devoted to sharing analyzable documents (e. g., text), the text processing techniques can be applied to create peers summaries. In this case, offered files and accordingly peers content can be represented in vector form (e. g., [29]).

Every time a query is forwarded, a forwarding algorithm will consult these summaries to select the neighbor to be contacted. If Bloom filter is used, the query keywords are hashed by using the same hash functions which were used to calculate Bloom filter. The obtained values are used to check keywords membership in a Bloom filter. In case analyzable documents are used, a query is matched with vector representation of peers content as it was suggested in [29].

In both cases, a neighbor which is supposed to store required content is contacted.

Content-Oriented Structure Building

In contrast to chaotically formed network structure, it is proposed to build connections between peers based on relations between respective users. Those relations in P2P file sharing applications are described and determined by shared (offered) and requested files. For example, if users offer similar content then there is a relation between them. The locally offered content as well as requested content is analyzed to create respective summaries.

Each peer analyzes summaries of its neighbors and restructures its neighborhood.

The experiments have demonstrated [28] that the best system performance was achieved when neighborhood includes two types of neighbors: those which offer the most similar content to the one shared locally (so-called *social neighbors*) and those which offer the most similar content to the one a user is interested in (so-called *egoistic neighbors*). In this case, the resulting global structure is content-oriented which represents grouping of peers based on not only similar offered content, but also users' interests.

Each peer has a possibility to collect information about other peers in a network which are not direct neighbors. To avoid additional messages for network structuring purposes, it was proposed to combine network discovery with a search process. In this way, a search message includes additional data-Log which contains a list of visited peers and summaries of their content. When a search message comes back to the peer-search initiator, it brings a list of visited peers[1]. A peer analyzes a message Log and current neighbors and selects the most appropriate peers for the new neighborhood.

The content-oriented structure building requires neither global control nor complete knowledge for its operations. All algorithms are locally executed. The global structure emerges as a result of individual activities of peers even under the highly dynamic conditions.

The resulting structure improves search operations, since at the beginning, egoistic neighbors are used to forward a query to the peers which might offer required content and then social neighbors are used to forward a message within a group offering similar content.

One can build an analogy with WWW. To search a page, a user can use its bookmarks to go as close as possible to the required content and then it can use the links provided on that page to other pages offering content about the similar topic.

4 Conclusion

The article presented a survey of existing P2P protocols. It was proposed to categorize the protocols depending on how they search data and organize the network structure.

The presented survey has demonstrated that the network structure and the respective forwarding algorithms significantly influence the properties of P2P applications. The existing systems are mainly unstructured so that P2P protocols mostly employ flooding which results to the traffic problem. On the other hand, the application of highly structured systems is limited due to high dynamics of network population.

References

1. Shirky, C.: What is P2P ...and what isn't? The O'Reilly Network. http://www.openp2p.com/pub/a/p2p/2000/11/24/shirky1-whatisp2p.html (2000)
2. Daswani, N., Garcia-Molina, H., Yang, B.: Open problems in data-sharing peer-to-peer systems. In: Database Theory-ICDT 2003, 9th International Conference. Volume 2572 of Lecture Notes in Computer Science., Siena, Italy, Springer (2003) 232–241
3. Internet2: Weekly reports. http://netflow.internet2.edu/weekly/ (2003)
4. Ripeanu, M., Foster, I.: Mapping the Gnutella network: Macroscopic properties of large-scale peer-to-peer systems. In: Peer-to-Peer Systems, First International Workshop, IPTPS 2002, Revised Papers. Volume 2429 of Lecture Notes in Computer Science., Cambridge, MA, USA, Springer-Verlag, Berlin (2002) 85–93
5. Gnutella: http://www.gnutellanews.com (2002)
6. from http://www.LimeWire.com, A.: The Gnutella protocol specification v0.4. (2003)

[1] In the current version of algorithms. For practical realization, a peer can analyze passing messages, each carrying the content summary of its starting peer

7. Wikipedia: FastTrack. http://en.wikipedia.org/wiki/FastTrack (2004)
8. Kazaa: http://www.kazaa.com (2003)
9. Leibowitz, N., Ripeanu, M., Wierzbicki, A.: Deconstructing the Kazaa network. In: 3rd IEEE Workshop on Internet Applications (WIAPP'03), San Jose, CA (2003) 112–119
10. Lv, C., Cao, P., Cohen, E., Li, K., Shenker, S.: Search and replication in unstructured peer-to-peer networks. In: ACM, SIGMETRICS 2002. (2002)
11. Adamic, L., Huberman, B., Lukose, R., Puniyani, A.: Search in power law networks. Physical Review **E 64 (2001)** (2001) 46135–46143
12. Deo, N., Gupta, P.: World Wide Web: A graph-theoretic perspective. Technical Report CS-TR-01-001, School of Computer Science, University of Central Florida, Orlando, FL 32816, USA (2001)
13. Unger, H., Wulff, M.: Cluster-building in p2p-community networks. In Akl, S., Gonzalez, T., eds.: Parallel and Distributed Computing and Systems, Cambridge, USA, ACTA Press (2002) 685–690
14. Jain, A.K., Dube, R.C.: Algorithms for Clustering Data. Prentice Hall (1988)
15. Krishnamurthy, B., Wang, J., Xie, Y.: Early measurements of a cluster-based architecture for P2P systems. In: ACM SIGCOMM Internet Measurement Workshop, San Francisco, USA (2001)
16. Rohrs, C.: Query routing for the Gnutella network. http://rfc-gnutella.sourceforge.net (2001)
17. Bloom, B.: Space/time trade-offs in hash coding with allowable errors. Communication of ACM **13(7)** (1970) 422–426
18. Crespo, A., Garcia-Molina, H.: Routing indices for peer-to-peer systems. In: 22 nd International Conference on Distributed Computing Systems (ICDCS'02), Vienna, Austria (2002) 23–33
19. Tsoumakos, D., Roussopoulos, N.: Adaptive probabilistic search for peer-to-peer networks. In: Third International Conference on Peer-to-Peer Computing (P2P'03), Linköping, Sweden (2003) 102–110
20. Tsoumakos, D., Roussopoulos, N.: A comparison of peer-to-peer search methods. In: Sixth International Workshop on the Web and Databases., San Diego, USA (2003)
21. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S.: A scalable content addressable network. In: Proceedings of ACM SIGCOMM 2001. (2001)
22. Stoica, I., Morris, R., Karger, D., Kaashoek, F., Balakrishnan, H.: Chord: A scalable Peer-To-Peer lookup service for internet applications. In: Proceedings of ACM SIGCOMM 2001. (2001) 149–160
23. Rowstron, A., Druschel, P.: Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In: IFIP/ACM International Conference on Distributed Systems Platforms (Middleware). (2001) 329–350
24. Zhao, B.Y., Kubiatowicz, J.D., Joseph, A.D.: Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Technical Report UCB/CSD-01-1141, UC Berkeley (2001)
25. Balakrishnan, H., Kaashoek, M.F., Karger, D., Morris, R., Stoica, I.: Looking up data in p2p systems. Communications of the ACM **46** (2003) 43–48
26. Ratnasamy, S., Shenker, S., Stoica, I.: Routing algorithms for DHTs: Some open questions. In: Peer-to-Peer Systems, First International Workshop, IPTPS 2002, Revised Papers. Volume 2429 of Lecture Notes in Computer Science., Cambridge, MA, USA, Springer-Verlag, Berlin (2002) 45–51

27. Chu, J., Labonte, K., Levine, B.: Availability and locality measurements of peer-to-peer file systems. In: SPIE ITCom: Scalability and Traffic Control in IP Networks. Volume 4868. (2002)
28. Sakaryan, G., Unger, H., Lechner, U.: About the value of virtual communities in P2P networks. In: 4th IEEE International Symposium and School on Advanced Distributed Systems (ISSADS 2004), Guadalajara,Mexico (2004)
29. Sakaryan, G., Unger, H.: Self-organization in peer to peer communitites. Technical Report UR-TR-0403, University of Rostock, Computer Science Dept. (2003)