

# Peer to peer networks: sharing between peers

Trond Aspelund

## Abstract

In this literature survey we look at peer-to-peer networks. We first see how peer-to-peer networks distinguish themselves from the client/server service model. Then we go more into the detail on how searches are done in the two different forms of peer-to-peer networks – Hybrid and Pure peer-to-peer, and look at some experiments with different search algorithms. Another important factor is the amount of network resources used to transfer the data, the overhead involved in maintenance and the cost of searching the peer-to-peer network. At the end we round up with some important security issues with peer-to-peer and some solutions to them.

## 1 Introduction

Peer-to-peer networks have become a large user of the Internets resources. They provide improved robustness, scalability and diversity over the standard client/server model by using methods that reduce the need of central servers to transfer data among the users [1]. An important part of the peer-to-peer networks is the search method used to find data among the peers in such networks. As the data is no longer located at a central server, other methods have to be used to find data. The large amount of data transferred in these networks uses much of the resources of the ISPs and organizations.

The words node and peer are used interchangeably, this because of the way it is used in the articles and in the definitions of the protocols and networks.

## 2 Network forms

Most of the traffic on the Internet has been based on the client/server model where a client connects to a server to retrieve data. Over the last few years more and more users have begun using the peer to peer networks where there are no distinction between the server and the client [2]. Figure 1 shows three network forms, client/server (a), Hybrid peer-to-peer (b) and Pure peer-to-peer (c). Communications are represented with arrows. The client/server model has been the most used method for sharing files on the Internet. WWW and ftp are well known services of this type. In this model the client initiates the communication and the server sends the file to the client or the client uploads the file to the central server [2]. Hybrid peer-to-peer network also uses a central server but only to control the list of clients and to do lookups among other user's files [3]. The files themselves are transferred directly between the clients without any mediation of the server. The server in this case can consist of several



## Hybrid peer-to-peer searching

In this model the client uploads its list of files to the server. This can be done in two different ways, batch or incremental. In the batch method the user uploads the metadata of the library on connection to the server and the data is removed when the user disconnects. In incremental update of client lists the metadata of the library is kept on the server and when the user rejoins only changes are updated. [4] presents four ways to distribute the metadata among the servers, chained, unchained, full replication and hash. In chained mode each server tries to satisfy the search locally, but if it can't find enough results it will forward the query to a remote server. Unchained has no communication among the server so only local metadata is used. Full replication architecture means that all servers receives all metadata from all servers and do the query locally. Hash divides the servers by distributing the hashed metadata words to different servers, so each server holds a subset of the metadata [4].

Yang et al [4] looks at which of the architectures is the best strategy and concludes with that chained is the best suited as it's fast, scalable and require the least amount of memory. Full replication demands more memory as each server need to hold all the metadata that can become very big with many users. Hash has very high bandwidth requirements and is best suited for systems with little exchange of metadata. As most searches are done within a small amount of nodes the searches can be efficient with short delays [4].

## Pure peer-to-peer searching

An important aspect of Pure peer-to-peer networks is how to spread queries among the peers. Two common methods are the depth first (DFS) and breadth first algorithms (BFS). Freenet, Chord and CAN are examples of the former and Gnutella the later [10].

Mache et al [10] looks at improvements on the search algorithm in Freenet. By using failed requests to add reference to the routing table of nodes, they improved the search results from the original algorithm. By adding only 25% of the reference it also performed better after a large amount of actions [10]. They also combined this with a breadth neighbour algorithm to even improve the results. This prioritizes the neighbour nodes. These improvements contributed by a factor of up to 9.25. Menasc et al [11] looks at implementing of a directory in each node that caches data of searches that comes through and a probabilistic message dissemination method to prevent the flooding of information that exists in other peer-to-peer networks like Gnutella.

Yang et al [12] looks at three ways to improve the search methods used in Gnutella, Iterative Deepening, Directed BFS and Local Indices. Iterative Deepening is based on using several queries with different depths. First it will use the lowest depth, if that is unsuccessful, it will send a resend message with TTL of the first search, preventing that nodes closer than the first depth will do any lookup for the search again. The node at the first depth will then continue the search until next depth is reached and the procedure will repeat itself for number of iterations. Directed BFS will use statistics of nodes that gives good results and only transmit to these nodes reducing the number of nodes to transmit to. Nodes that receive a query use normal BFS search. In a local indices architecture nodes cache metadata for nodes within a given radius. When a node joins it

will transmit its metadata to adjacent nodes. When a search is performed it will only be processed at nodes given in a policy for the system. The policy usually lists at which hop to perform searches and when the query is to be dropped. All these techniques reduce the aggregated cost of a search. A combination of BFS and DFS is tried in [13] where they look at using peer-to-peer in a information sharing system where there exists more topics than in a regular file sharing system. Their method exchanges the TTL with a number of nodes to visit, this number gets decremented in each node and split among the sub nodes when it gets transmitted further. Their experiments show that this approach performs better than the BFS algorithm but that the benefit gets smaller with a large amount of peers. Their algorithm makes better use of the “small world phenomena” and works better when there is a change of interest among the peers i.e. looking for other files.

Another way to do searches in a peer-to-peer system is using a P-Grid approach. P-Grid makes use of a virtual binary search tree and the prefix of the search key [14]. By dividing the keys up by the use of binary tries each host has a list of prefixes and peers that can fulfil that part of the search. So if it can't fulfil the query by itself it sends it off to the peer in its list with the longest matching prefix. When two peers meet they will divide the prefix between them, if they share a common prefix they can initiate new exchanges by forwarding each other to peers in their reference list. If they are in a prefix relationship the one with the shorter key can specialize and extend its prefix [14] [15].

As this uses a normal binary search method it reduces the number of messages and keeps them quite constant even for a great number of peers. The approach also has redundancy in the way that several nodes hold the same prefix so if one peer disappears the information stored by that peer will still exist in the system.

Randomness is used where the search is dispatched to prevent queries to always end up in the same group of nodes as all information in that prefix might be divided among nodes in different parts of the tree. This happens because how the information is updated depends on the interconnection of the peers. Simulations on the number of messages [15] shows that Gridtella (a Gnutella compatible peer-to-peer system using P-Grid) uses only 61-72 messages on systems consisting of 20000-200000 peers compared to Gnutella using 8744-78728 messages. The number of messages on Gnutella doesn't show any increase when there are more than 80000 peers. On the other hand a comparison between Freenet and P-Grid shows that both use about the same number of queries when the network is stable [16]. Freenet needs to use a large amount of messages until the network is stable while P-Grid uses few messages even before it is stable. As a comparison, P-Grid has a much smaller routing table than Freenet. P-Grid uses 35 entries compared to Freenet's 250. When the number of routing table entries for Freenet is reduced to 35, the amount of messages increases to a much higher number, showing that Freenet's method relies on a large number of routing entries.

## 4 Bandwidth usage

### Overhead

A peer-to-peer network will base its communication on the flow of the protocol and doesn't use the topology of the routers for its signalling. As all traffic gets routed internally in each node, traffic between nodes relies on the normal Internet topology. This may, among other things, make the queries pass the same router several times during a query depending on the location of the nodes in the network. The signalling by itself, like keep-alive messages does also create much traffic. A measurement done by Matei [17] in November 2000 on the Gnutella network showed that only 35 percent of the traffic was queries and the rest was overhead. The remaining 55 percent was maintenance of group memberships. Measurements done in June 2001 showed that this was improved in newer version as the queries now used 91 percent of the traffic. They also made an estimate for a large Gnutella network of 50000 nodes that shows that it would use about 1GBps of bandwidth excluding file transfers. This adds up to about 330 Tbytes per month which is 1.7 percent of the total traffic estimate for the U.S. Internet backbone in December 2000.

### Requests vs. Bytes

On a measurement done on Kazza [9] they divided the size of the requested objects in three groups, less than 10MB, between 10 and 100 MB and above 100MB. These measurements showed that the largest number of request were done for small files (91%) but most of the traffic (65%) is for the files larger than 100MB.

### Locality

Sen and Wang et al [18] measure the traffic on the border routers of a large ISP using the known ports for three well-known peer-to-peer network, Fast-track (used by Kazza), Gnutella and Direct Connect. This method made the measurement see all the traffic without differentiating between signaling and actual data that were transferred. This approach also has the benefit that it doesn't care if the data is encrypted or the need for any knowledge over the protocols. On the other hand it won't see any data that is transferred on other non standard ports. For all the different peer-to-peer networks the measurements show that the amount data transferred is not spread evenly among the nodes, but rather that a small amount of nodes is transferring a large amount of the data, the top 1 percent of the IPs transferred between 64-73 percent of the data. It seems also, that a few hosts have a large amount of connections to other hosts. This is somewhat better when you look at the prefixes or the Autonomous Systems, which shows communication with a larger group. The high volumes and good stability of this kind of traffic can give the ISPs the possibility to use application-specific layer-3 traffic engineering to manage the workload this kind of traffic generates in their networks [18].

Sen et al [18] and Gummadi et al [9] suggests that Indexing/Caching might be a method to reduce the amount of data that is transferred in and out of an organisation or ISP. Also the use of locality aware search mechanisms would help reduce the load created by this kind of traffic. In Kazaa a large percentage of the

data (86%) could be avoided using a proxy [9]. The use of Indexing/Caching mechanisms is a solution very few organizations want to use because of legal and political situations that could arise if they stored illegal content [9]. Lui et al [2] looks at the interoperability between peer-to-peer networks by the use of peer-to-peer gateways to translate between the different protocols; this should also be a good place to put any such caching functionality but will have the same legal problems as a pure caching mechanism.

## 5 Security

Security requirements in a peer-to-peer system can be assigned into four general areas – availability, file authenticity, anonymity and access control [19]. Nodes might be attacked (Denial of service), files are not what they say they are, preventing people from seeing who shares what or users may share private files without their knowledge.

Many things can reduce the availability in a peer-to-peer network. A malicious peer might join the network giving false replies to queries to make other peers download viruses or other junk files or give replies to all queries that point toward one host to make a denial of service attack against that peer [1][19]. Other methods might be to serve files at low speeds so downloads takes ages preventing the peer to get the file it tries to retrieve.

Even in a network without any malicious peers search results might yield conflicting results. A search for a given keyword returns a number of results that looks like the same but are different. Which of the results is then the most authentic? Prioritizing of the results could be done on the basis of age of object, graded by experts, voting on the object or reputation of the one giving the results [19].

There can be many reasons for the need of anonymity like censorship resistance, freedom of speech, sharing of copyrighted material, protection of privacy and retrieval of public information in a police investigation [19].

Another problem is how to manage access control in an open peer-to-peer system so that no secret or private documents or copyrighted material are shared [19].

A method to improve the availability and file authenticity is to add a reputation system to the network that users can use to vote after they downloaded a file. These methods must have ways to prevent misuse of the system to gain better reputation and ways to hold this kind of information in a peer-to-peer system where peers joins and leaves at random [1].

A method to implement access control is to have secure peer-groups in the peer-to-peer network that authenticates users with passwords or domains and prevents the sharing of files with certain rights outside that group [20]. Also how the users add files for sharing is a problem as novice users might share more files than they think they do [21] or different persons do different things on the computer like one storing private files while another shares files without looking or caring about what is shared [21].

Anonymity also creates a problem, first you need to locate what you want to download and then download it. But then if you dont know where the file lies and cant get hold on that information how do you then download it? Some try to create server anonymity through broadcast searches (Free Haven), others

don't have server anonymity but author anonymity (Freenet). A way to have anonymity is to use a series of proxies to communicate among the peers so no one communicates directly with the source of the information, this can be done both ways to prevent anyone for knowing who they really talk to. The proxies may be other clients in the network [19]. These kinds of proxies might reduce the performance of the search on the peer-to-peer network.

## 6 Conclusion

Peer-to-peer networks produce a large amount of traffic in today's Internet. One important part of a peer-to-peer network is its search algorithm. The search method used has a great impact on how many messages are used to find the items in the query. The location of the nodes will also have impact on the load these networks generate on the infrastructure. Pure peer-to-peer networks is also moving toward having nodes working as a central meeting ground to get some of the architectural benefits from the Hybrid peer-to-peer networks but keeping the availability and robustness benefits of not predefining any nodes to this task but have the network do it dynamically. By the use of proxies or by implementing search algorithms that take locality into account, the resources usage on the Internet could be reduced. Security is also an important part that has not been addressed by many peer-to-peer networks but more and more start to think about how to improve the anonymity of the users and ways to implement methods to create ways to use peer-to-peer to share secret information within a group.

## References

- [1] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," in *Proceedings of the twelfth international conference on World Wide Web*. ACM Press, 2003, pp. 640–651.
- [2] S. M. Lui and S. H. Kwok, "Interoperability of peer-to-peer file sharing protocols," *SIGecom Exch.*, vol. 3, no. 3, pp. 25–33, 2002.
- [3] Y. Itakura, M. Yokozawa, and T. Shinohara, "Model analysis of digital copyright piracy on p2p networks," in *Applications and the Internet Workshops, 2004. SAINT 2004 Workshops. 2004 International Symposium on*, Jan. 2004, pp. 74–79.
- [4] B. Yang and H. Garcia-Molina, "Comparing hybrid peer-to-peer systems," in *Proceedings of the 27th VLDB Conference, Roma, Italy*, 2001.
- [5] "Napster home page," May 2004. [Online]. Available: <http://www.napster.com/>
- [6] "Winmx home page," May 2004. [Online]. Available: <http://www.winmx.com/>
- [7] "Gnutella home page," May 2004. [Online]. Available: <http://www.gnutella.com/>

- [8] "The freenet project," May 2004. [Online]. Available: <http://freenetproject.org/>
- [9] K. P. Gummadi, R. J. Dunn, S. Saroiu, S. D. Gribble, H. M. Levy, and J. Zahorjan, "Measurement, modeling, and analysis of a peer-to-peer file-sharing workload," in *Proceedings of the nineteenth ACM symposium on Operating systems principles*. ACM Press, 2003, pp. 314–329.
- [10] J. Mache, M. Gilbert, J. Guchereau, J. Lesh, F. Ramli, and M. Wilkinson, "Request algorithms in freenet-style peer-to-peer systems," in *2nd International Conference on Peer-to-Peer Computing (P2P 2002), 5-7 September 2002, Linköping, Sweden*. IEEE Computer Society, 2002, pp. 90–95.
- [11] D. A. Menasc and L. Kanchanapalli, "Probabilistic scalable p2p resource location services," *SIGMETRICS Perform. Eval. Rev.*, vol. 30, no. 2, pp. 48–58, 2002.
- [12] B. Yang and H. Garcia-Molina, "Improving search in peer-to-peer networks," in *Proceedings of the 22 nd International Conference on Distributed Computing Systems (ICDCS'02)*. IEEE Computer Society, 2002, p. 5.
- [13] Y. Ren, C. Sha, W. Qian, A. Zhou, B. C. Ooi, and K.-L. Tan, "Explore the "small world phenomena" in pure p2p information sharing systems," in *3rd International Symposium on Cluster Computing and the Grid*, May 2003, pp. 232–239.
- [14] K. Aberer, "P-Grid: A self-organizing access structure for P2P information systems," in *Sixth International Conference on Cooperative Information Systems (CoopIS 2001)*, 2001, <http://www.p-grid.org/Papers/CoopIS2001.pdf>.
- [15] K. Aberer, M. Puceva, M. Hauswirth, and R. Schmidt, "Improving data access in p2p systems," *Internet Computing, IEEE*, vol. 6, pp. 58–67, Jan./Feb. 2002, <http://www.p-grid.org/Papers/IC2002.pdf>.
- [16] K. Aberer, M. Hauswirth, and M. Puceva, "Self-organized construction of distributed access structures: A comparative evaluation of p-grid and freenet," in *In Workshop on Distributed Data and Structures, June 2003*, June 2003, <http://www.p-grid.org/Papers/TR-IC-2002-74.pdf>.
- [17] R. Matei, A. Iamnitchi, and P. Foster, "Mapping the gnutella network," *Internet Computing, IEEE*, vol. 6, pp. 50–57, Jan./Feb. 2002.
- [18] S. Sen and J. Wang, "Analyzing peer-to-peer traffic across large networks," vol. 12, pp. 219–232, Apr. 2004.
- [19] N. Daswani, H. Garcia-Molina, and B. Yang, "Open problems in data-sharing peer-to-peer systems," in *Proceedings of the 9th International Conference on Database Theory*. Springer-Verlag, 2002, pp. 1–15.
- [20] Z. Li, Y. Dong, and L. Z. and; Jianhua Huang, "Implementation of secure peer group in peer-to-peer network," in *Communication Technology Proceedings, 2003. ICCT 2003. International Conference on*, vol. 1, Apr. 2003, pp. 192–195.

- [21] N. S. Good and A. Krekelberg, "Usability and privacy: a study of kazaa p2p file-sharing," in *Proceedings of the conference on Human factors in computing systems*, 2003, pp. 137–144.