P2P searching methods, research issues, solutions and their comparison

R. Denneman
Faculty of Management and Governance
University of Twente, the Netherlands
rudo.denneman@gmail.com

ABSTRACT
An important feature that has to be fulfilled by Peer-to-Peer (p2p) applications is associated with searching files, contents and entities. This paper discusses a number of p2p searching methods and it derives certain criteria that can be used to compare these P2P searching methods. From this comparison it could be derived that the central index search method can be the most effective for p2p file sharing applications and the Distributed Hash Table search method for p2p telephony applications.

Keywords
Peer-to-peer, search method classes, requirements, qualitative comparison.

1. INTRODUCTION
Peer-to-peer (p2p) networks have great potential to relieve the workload of centralized servers in the traditional client-server model by utilizing bandwidth, storage space and processing power of all connected nodes.

One of the challenges in p2p networks is indexing and searching the content of nodes also known as routing. Centralized search indexes require vast indexes to store the content of all connected nodes. Local and decentralized indexes return incomplete results because their search queries die after several hops when exceeding their Time To Live (TTL) to avoid flooding the network [1]. Most contemporary p2p file sharing networks use decentralized indexes [2] but there are several other p2p applications like media streaming and VoIP that have different requirements for p2p search method classes.

[1] classifies p2p protocols on the search method and network structure. In the context of this search method a distinction can be done between random probing of peers known as blind search and peers that have indexed files of connected peers known as informed search. The network structures are divided in unorganized structures in which connections between peers are made chaotically and organized structures. These organized structures are subdivided. The first category contains protocols that rely on a two layer structure with super nodes, peers that have more computational power and bandwidth available, and "weaker" peers that connect to them. The second category contains protocols that connect peers using data placement. Moreover, [1] continues to discuss different combinations of the previously mentioned categories and concludes that most p2p applications today use unstructured networks and use blind flooding search which results in high bandwidth usage.

[3] gives an overview of influential work in p2p content search. It discriminates between text-based search and content-based search methods as well as centralized and decentralized networks. Centralized networks index the content of all nodes while decentralized networks tend to have clients search their immediate peers using packets with a TTL value. This paper also compares the legal issues of the different search methods. It also notes that more and more p2p systems extend their search capabilities, for example some networks now also support proximity queries, wildcard queries and phrase queries. Content-based search methods supported by contemporary p2p networks are limited. Although some networks support keywords, others support author names or descriptions for documents.

Another survey of p2p search methods [2] provides an overview of p2p search methods and their dependability or robustness. It concludes that most contemporary p2p networks have been organized (structured) to aid in searching for content. The organization is either done by maintaining indexes at super peers, also called super nodes, or by Distributed Hash Tables (DHT). A distinction is also made between semantic-free indexes, like DHT, and semantic indexes. Semantic indexes store more information about an object than just the key that is required to find it using DHT. Keywords or other aspects describing the object can be used in search methods. It also states that in order for p2p to be used for more applications besides file sharing, robustness is essential.

While these surveys listed above focus on the different p2p search methods and p2p networks structures, no comparison is given between these p2p search methods, using predefined criteria.

This paper will compare different p2p search method classes and assess their ability to satisfy the search requirements imposed by main p2p applications, e.g., file sharing, p2p telephony.

The main research question of this paper is:

- Can existing p2p searching methods satisfy the requirements imposed by main p2p applications?

  In order to answer this main research question a number of sub questions are derived:

  1) What are the available p2p search method classifications?
  2) What are the main p2p application classes?
  3) What search requirements are imposed by p2p applications on p2p systems?

11th Twente Student Conference on IT, Enschede, June 29th, 2009

4) Are the available p2p search method classes able to satisfy these requirements?

This research is a literature study and qualitative comparison of several p2p search method classes. Since much research has already been done in the area of p2p searching, a literature study is sufficient to find the information needed for a qualitative comparison.

This paper is organized as follows. Section 2, answers the first research sub-question, and it describes the p2p search method classes and gives examples of implemented protocols. Section 3, answers the second research sub-question and it lists several p2p application classes and identifies two main p2p application classes. Section 4 discusses the search requirements imposed by these p2p application classes. Moreover, section 4, answers the third sub-question. A qualitative comparison of the search method classes based on their ability to satisfy these requirements is provided in section 5. Section 5, answers the fourth research sub-question. Finally the conclusion and future work are given in section 6.

2. SEARCH METHOD CLASSES

In [4] the search methods are divided into four classes namely centralized index, query flooding, hierarchical overlay and distributed hash tables (DHT). In this section each of these classes will be described along with applications which use these classes.

2.1 Centralized index

The search method in these networks is actually based on the client server model. Each node sends an update with its content to the centralized server on joining the network or when its content is updated. This setup is shown in Figure 1. The centralized server keeps track of all node content and answers search queries from nodes. The problems associated with this search method are, see [4]:

- Single point of failure: A crash of the central index paralyzes the entire network since no searches can be initiated.
- Scalability: Expenses of the centralized index are high when the network expands. This can cause searches to be executed slowly.
- Legal issues: In some countries the owners of centralized indexes of illegal content can be prosecuted.

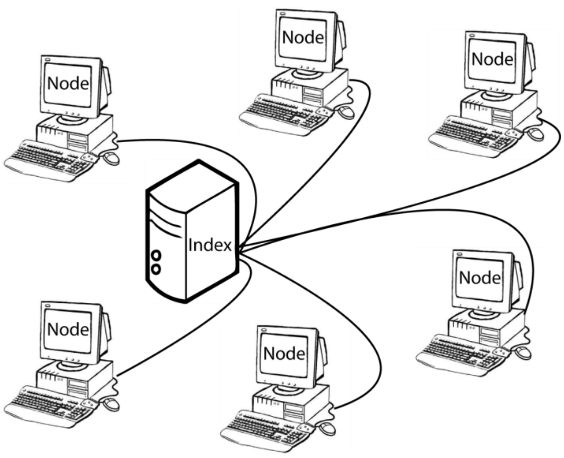

Figure 1: Search through a centralized index

2.1.1 eDonkey2000

The eDonkey2000 application, see [5], is very successful, since more than 4 million users could be online using this p2p applcaition. This p2p application uses multiple servers but the setup is clearly using a centralized server system. The server indexes the client content and keeps track of which other centralized servers exist, but does not exchange search queries with the other servers. Servers provide the list with all online servers to connecting clients. Since there are between 100 and 200 servers on the list the client node always has several options to connect to when it is started. Lists of available servers are also available online, thereby solving the initial connect or bootstrap problem [5].

2.1.2 Bittorrent

Another p2p application that uses a centralized server is Bittorrent. [5] Although the Bittorrrent application doesn't provide search functionality of its own but works by indexing .torrent files on a website. These files provide the client with an URL to the torrent tracker. A tracker is a server that keeps track of several randomly selected peers that are downloading and/or uploading the shared content. A peer connecting to the tracker will connect through these peers and then to other peers that are downloading or uploading the content, this collection of peers is called a swarm. The high speed of downloads have made this type of p2p file sharing very popular. This search method does however sacrifice the anonymity of the peers [5].

2.2 Query flooding

Query flooding operates by constructing an overlay network between nodes. In this overlay network nodes are connected to a small number of neighbors. When one of the nodes initiates a search query ("Node 0" in Figure 2) it will send this query to each of its neighbors. The neighbor will check its own file index and forward the query to its own neighbors thereby extending the range of the search. When the requested file is found the results will travel back to the initiating node via the same route. Since there is no central server present in this network structure, nodes need to find a way to initiate contact with other peers. One solution for this is for a node to maintain a list of nodes that are often available [4].

The flooding of queries has a significant impact on network traffic limiting its scalability. One way to diminish this effect is to equip every query with a TTL value which decreases on every node it passes, halting the query when TTL has reached 0. Using TTL in query flooding on large networks does however impact the completeness of the search since the query will not reach all connected nodes [6].

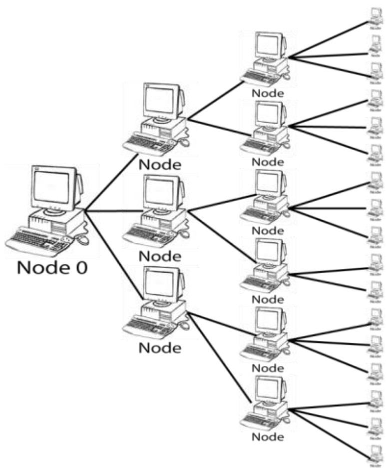

Figure 2: Search through query flooding

2.2.1 *Ad hoc self supervising networks*

[7] suggests a way to increase efficiency of searching within query flooding networks by allowing nodes to connect to other nodes and breaking connections when other nodes are sending them too much traffic. This enables the network to self regulate and avoids overloading nodes, thereby making the network more stable. One advantage over a network with super nodes, see section 2.3, is that this technique also works when there are no nodes that have sufficient resources to take on the role of super node.

2.2.2 *Unstructured adaptive lookup*

Adaptive indexing works by having each node store the bandwidth and delay of the connection to its neighbors. When a query is initiated the user can specify certain requirements for the connection latency and bandwidth. The query is then sent to the neighbors which satisfy these connection requirements. Each neighbor then forwards the query to its neighbors that satisfy the connection requirements and so on, until the TTL value for the query reaches zero. The result of this method is less network load intensive, due to the use of the selective forwarding, see [8].

2.3 Hierarchical overlay

In the hierarchical overlay design centralized index and query flooding are combined. According to [4] the network is constructed out of normal nodes (leaf nodes) and super nodes. A super node being a node with higher bandwidth and computational resources than leaf nodes. The super node has several leaf or child nodes of which it indexes the content. This setup is shown in Figure 3. Each super node has several normal nodes as children. The super node builds and index of the content of its children. When the requested content isn't found among the children of the super node, the super node starts a search using query flooding to other super nodes to which it is connected. Super nodes are typically more powerful nodes that have more bandwidth at their discretion.

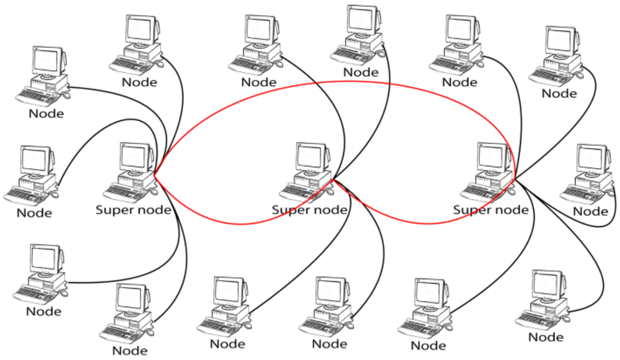

Figure 3: Search in a hierarchical overlay network

2.3.1 *Skype p2p internet telephony protocol*

Skype is a popular p2p telephony client. The Skype network consists of (leaf) nodes, super nodes and a central login server. Upon connecting to the network a node must connect to a super node and the central login server. To find a super node to connect to, the Skype protocol, stores the IP numbers and ports of super nodes in a cache at each host. The cache is updated regularly. When all of these super nodes are unreachable, the node contacts one out of seven IPs that are hardcoded in the executable [9].

Although the Skype protocol is proprietary an analysis done in [9] suggests that a Skype node contacts the super node when the search is initiated. The super node then returns the IP and port number of eight nodes which the client then queries. When the user is not found the node contacts the super node again and receives 16 additional node addresses to query. This process continues until the user is found or the node gives up. Before the node gives up it will query the Skype login server. The login server is believed to be a fall back option when the regular search method fails to check whether the username exists.

2.3.2 *FastTrack protocol*

The FastTrack protocol [10] was one of the first p2p protocols that used the differences in node resources to create a hierarchical overlay in its network. The FastTrack protocol operates by having the super nodes index the content of the leaf nodes connected to it. This protocol also creates a content hash of each file in order to identify similar files, allowing downloading from multiple sources.

When a node sends a query to a super node, the super node replies with the IP addresses and ports of the nodes which have files, matching the keywords as well as the metadata found for these files. In order to do this, the super node maintains TCP connections to other super nodes to which it forwards the query using query flooding [10].

It must be noted that the list of super nodes is provided by a central index server. This server also provides the login service [5].

2.4 Distributed hash tables

The DHT design enables searching p2p networks in a fully distributed manner. [11] describes the DHT algorithm, it requires that the data can be identified using a unique numeric hash key and that nodes store each other's numeric keys. The DHT algorithm implements just one method. Given a key this method returns the node that stores the data corresponding to the numeric key.

In a DHT network all nodes and data is identified by a number. The keys are stored at one or more nodes that have identifier number which is close to the stored key. This way nodes can guarantee that a query for identifier id will only be forwarded to a node closer to the id. Eventually the query will arrive at the closest node. There are different implementations for this algorithm e.g. Chord [12], CAN [13] and Pastry [14] but the principle remains the same. Each node also keeps track of some other nodes using a routing table to be able to use the search algorithm [5].

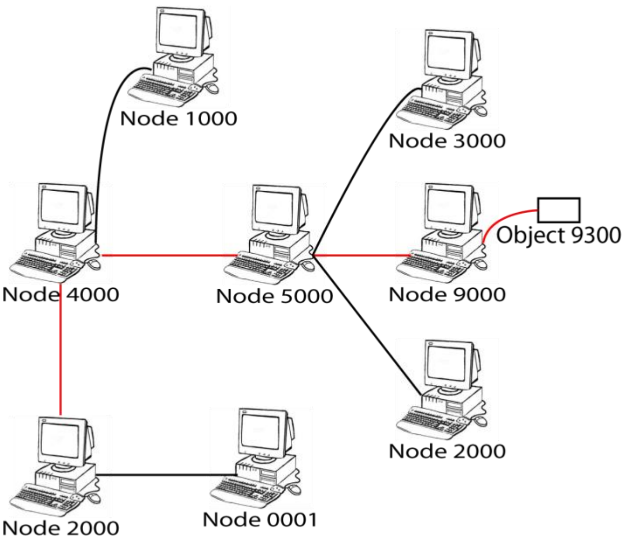

Figure 4: Search in a DHT network

For example, in Figure 4 consider that node 2000 is searching for object 9300. Consider that it has knowledge only of the 2 following nodes, 0001 and 4000. Since 4000 is closer to 9300 than 0001 it will send the query to 4000. Using the same principle node 4000 will forward the query to 5000 which will then forward it to 9000. 9000 is closest to the object and responsible for its storage.

2.4.1 *Session Initiation Protocol using DHT*

[15] suggests an implementation of the IETF's session Initiation Protocol (SIP) [16] with DHT for p2p telephony purposes. The DHT implementation used is an adaptation of the Chord protocol. It's adapted to use a hierarchical overlay consisting of leaf nodes and super nodes. The reason for this setup is that it enables NAT (Network Address Translation) traversal for regular nodes via the super nodes. A node joins the network by sending the SIP REGISTER multicast to the local network. In case no nodes are located in this manner the Service Location Protocol (SLP) or preconfigured bootstrap nodes can be used. Upon discovery of a node the connecting node will cache its address for future use. The node then sends a query with its identifier to receive its successor and predecessor in the Chord ring. It then sends an update query to its successor and predecessor so they can update their data structures. Similar to the previously mentioned Skype protocol the super nodes perform lookups, in this case using DHT, for leaf nodes. This implies that only super nodes are part of the DHT network. If the node initiating a lookup is a super node it will simply perform the DHT lookup itself. Leaf nodes simply connect to a super node without joining the DHT network.

2.4.2 *Selective publishing*

Due to DHT's large overhead its suitability for the distribution of popular files on a p2p file sharing network is questionable. [17] however makes the case for a hybrid solution between query flooding and DHT. A network is suggested in which popular items that can be easily located by query flooding are located that way and rare (non-popular) items are indexed in a DHT. Since popular items are easy to locate without a high TTL value this is preferred over DHT because DHT networks will have to be restructured with every joining or leaving node. Since the average connection time of leaf nodes and super nodes on the Gnutella network measured in this study was 58 minutes and 93 minutes, respectively, it means that the DHT will have to be restructured often causing more overhead. Popular items produce significantly more overhead when building the DHT index than rare (non-popular) items. This hybrid structure selectively publishes the items based on its popularity to the DHT or using query flooding.

2.4.3 *P2PSIP*

The IETF P2PSIP work group [18] is continually working on a P2P adaptation of the IETF's SIP [16] standardization. The goal is to create a protocol that can be used in combination with any DHT structure. The difference with the original SIP protocol is that establishing and managing sessions is no longer done by a central server but by multiple peers in the network. In this case, a peer functions as the previously described super node. It is important to emphasize that what previously was denoted as a leaf node or a normal node, it is called within the P2PSIP context as a client. The peers are registered in the DHT overlay structure but clients are simply connected to the peers use the resource discovery capabilities of the peers instead of querying themselves. This process is identical to that in the hybrid structure. In the hybrid structure there are large numbers of leaf nodes. In P2PSIP, the role of client is somewhat different and should only be fulfilled by devices that have significantly less resources than peers, like smart phones or other mobile devices. Another difference between P2PSIP and the original SIP protocols is that in P2PSIP NAT traversal and resource discovery are managed by the peers instead of a central server [19]. A representation of the signaling within a P2PSIP network can be found in Figure 5.

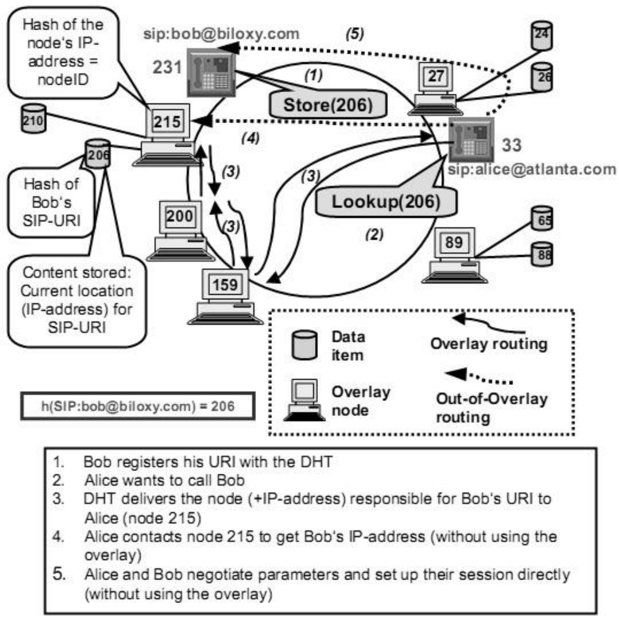

Figure 5: Schema of P2PSIP Signaling, from [20]

3. MAIN P2P APPLICATION CLASSES

There are many applications for p2p networking e.g.:

- Education and Academia: One initiative of the Pennsylvania State University is Lionshare. This p2p program is still under development but is designed to let users of connected educational institutions share knowledge [21].
- Military: p2p technology offers many benefits for the military such as direct sensor information sharing between combat units [22].
- Business: There are plenty of opportunities to use p2p applications for marketing purposes however there are ethical and legal issues that need to be solved, see [23].
- TV: [24] proposes a scalable VOD application based with the content distributed using p2p. Although the results are promising this type of video streaming application is not yet widely used.
- File Sharing: Perhaps this is the best known and popular p2p application. Exchanging files through p2p takes away the necessity for a central server thereby distributing the massive network load on a single server in the traditional client-server setup over a large number of nodes.
- Peer-to-peer telephony: P2P telephony allows its users from around the globe to communicate directly by providing an audio and in some cases video connection. A popular example is the Skype network. At the beginning of 2009 a milestone of 15 million online Skype users was reached [25]. In total the Skype network has over 443 million registered users [26].

Out of these applications, p2p file sharing and p2p telephony are the most widely used application classes. Therefore this paper will consider these as the main applications under study and examine their search requirements.

4. P2P APPLICATION SEARCH REQUIREMENTS

In this section the search requirements for the main p2p applications will be described. Their importance for the p2p application class will be addressed as well. A summary of the importance of search requirements for the different application classes is given in Table 1.

4.1 Scalability

Scalability is the ability of the search method to remain effective while the network is expanded.

In case of a file sharing p2p application this requirement's importance is rated as high. File sharing networks can attract millions of users [27] and therefore scalability is an important requirement for these networks.

Searching p2p telephony users directories also has to be scalable since the number of accounts can run into the millions. As mentioned before Skype alone has more than 443 million registered users [26].

4.2 Completeness of results

To which degree does the search method return complete results of all available data on the network.

For file sharing a list of all available files matching the query is desirable. However the nature of file sharing networks increases and the probability of the existence of multiple instances of a file on more than one node in the network also increases. This also increases the likeliness of a file being found even though the query has not reached all nodes in the network. For this reason the importance of this search requirement is rated as medium.

Since almost every user account in a p2p telephony network identifies a unique user completeness of results is essential. Otherwise users could be unable to locate one another.

4.3 Dependability

Does the search method cope with failing nodes? This requirement is also dependant on the required network structure.

In file sharing a failing node causes a loss of the data available on that node. Additionally, depending on the network structure, the node could also function as a super node in the overlay network. Without adequate restructuring of the network the children of the super node would be unable to initiate search queries or be found by other nodes. For this reason the importance of dependability in file sharing network is rated as high.

Dependability is also an important issue for search methods in p2p telephony. A failing centralized index cripples the entire network and a failing super node could cripple a branch of nodes.

4.4 Latency

How fast is a searching procedure completed?

For file sharing the importance of the latency requirement is rated as medium. Whether the user has to wait for ten seconds or two minutes is arbitrary. Refreshing the list of nodes that have the file that is being downloaded to enable simultaneous downloads can take a while as well.

In p2p telephony latency has a higher priority. When initiating a call the node has to look up the IP address of the receiving node. When this process takes too long it will seriously affect the user experience.

4.5 Security support

Does the solution provide security support?

Without security support file sharing networks can be vulnerable for DoS (Denial of Service) attacks. A super node could shut down a regular node by responding to all queries that the data is located on the target node. Another problem within p2p file sharing networks is file authenticity, the content of files could be altered or a file could be falsely named. Anonymity has also been raised as a security issue. Although most users would see anonymity in a file sharing network as a way to share illegal content without fear of prosecution, there are also other reasons for anonymity. For example to publish or access information in a hostile political climate [28].

Data stored by p2p telephony networks could include e-mail addresses, full user names and living addresses. To protect the privacy of users, search methods should be as secure as possible.

Message authenticity could also be a problem for routing. Malicious nodes could provide other nodes with bad routing information.

4.6 Use of standardized solutions

Is the p2p searching method using standardized protocols or not?

The importance of the use of standardized solutions has been rated as low for both p2p file sharing and p2p telephony. In both cases the most networks function autonomously and there is no harm in search methods being specifically designed for the application in which they are employed.

4.7 Mobility

Can the p2p searching method be used by both fixed and mobile users or only be fixed or only mobile users?

The increase in popularity of mobile networks and file sharing raise the importance of resource discovery on wireless networks although most p2p file sharing applications still target wired connections [29].

Also due to popularity of mobile networks and the increase of Wi-Fi capable smart phones the importance of use of p2p telephony on mobile devices is rated as high.

Requirement	File Sharing	Telephony
Scalability	High	High
Completeness of results	Medium	High
Dependability	High	High
Latency	Medium	High
Security support	High	High
Use of standardized solutions	Low	Low
Mobility	Medium	High

Table 1: Importance of requirements per p2p application class

5. QUALITATIVE COMPARISON

In this section the solutions provided in section 2 will be compared to the requirements provided in section 4 thereby answering the fourth sub question: “Are the available p2p search method classes able to satisfy these requirements?”. The solution’s ability to satisfy the imposed requirements is rated. The used ratings consist of Bad, Fair and Good grades. Furthermore the complete qualitative comparison results are listed in Table 2, in Appendix A.

5.1 Scalability

The grades Bad, Fair, Good can be interpreted for this criterion as follows:

Bad: Scaling is impossible or is limited by resources.

Fair: Scalable, but the performance of the search process decreases when the number of nodes increases.

Good: An increase of the number of nodes does not negatively affect the searching performance.

Some of the main P2P applications can be graded as follows:

- eDonkey2000: In practice the largest index has around 900,000 users. Scaling is limited by central server capacity [5]. Rating: *Bad*;
- Bittorrent: The trackers required to maintain a .torrent file only keep track of several peers not the entire swarm. This makes the protocol more efficient for trackers than a central index would be. The search process to find these trackers is handled outside of the protocol by central indexes in the form of websites. Rating: *Fair*;
- Ad hoc self supervising networks: Query flooding uses fully distributed indexes but generally suffers from a high network load. Scaling affects performance of the search process. Rating: *Fair*;
- Unstructured adaptive lookup: This method potentially limits the amount of nodes to which queries are flooded making the search algorithm more efficient than regular query flooding. Scaling affects performance of the search process Rating: *Fair*;
- Skype p2p internet telephony protocol: Traffic on network underlay links remains almost constant when the number of users increases. Scalability is restricted by the node / super node ratio. Super nodes are scarce this could impact the scalability [30]. The login server fallback option in the search algorithm could also be an issue. However, Skype regularly has 15 million users online simultaneously. Rating: *Good*;
- FastTrack protocol: Query flooding between super nodes causes high network load and decreases search performance [10]. Rating: *Fair*;
- Session Initiated protocol using DHT: Scalability of this protocol is estimated to be high but also depends on the number of super nodes and their resources [15]. Rating: *Good*;
- Selective publishing: This method has the same fair scalability as other query flooding techniques. The DHT addition to this protocol affects scalability as well. [17] tested the performance and creating the DHT index took 2-3 seconds per file. For a large number of files this would take a significant amount of time. Rating: *Fair*;
- P2PSIP: [31] shows that call setup latency is barely affected by the number of nodes but more dependent on the distance between nodes. Rating: *Good*;

5.2 Completeness of results

The grades *Bad*, *Fair*, *Good* can be interpreted for this criterion as follows:

Bad: Unpredictable results. Not knowing to what degree the network is searched is worse than knowing the limitations of the search protocol.

Fair: Popular items are almost always found. In most cases, e.g. in file sharing, popular items are the most important to be found however in the best base all results are returned.

Good: Results from the entire network are returned.

Some of the main P2P applications can be graded as follows:

- eDonkey2000: This network has a centralized index. Rating: *Good*;
- Bittorrent: This network has a centralized index. Rating: Good
- Ad hoc self supervising networks: Query flooding with a limited TTL value. No guarantee that all available files can be found. Rating: *Fair*;
- Unstructured adaptive lookup: Query flooding with a limited TTL value. No guarantee that all available files can be found. Rating: *Fair*;
- Skype p2p internet telephony protocol: All super nodes can access complete indexes of users. The login server acts as a back up to determine if a user exists. Rating: Good
- FastTrack protocol: The FastTrack also uses a query flooding technique with a limited TTL value. No guarantee that all available files can be found. Rating: *Fair*;
- Session Initiated protocol using DHT: Super nodes are connected to the DHT with the leaf nodes as their items. This ensure complete recall. Rating: *Good*;
- Selective publishing: Although theoretically this method has a higher recall rate than regular query flooding [17]. There is no guarantee items are always returned. Rating: *Fair*;
- P2PSIP: The DHT networks ensure complete recall. Rating: *Good;*

5.3 Dependability

The grades Bad, Fair, Good can be interpreted for this criterion as follows:

Bad: Leaving or failing nodes are not detected and cause the search process to fail or there is a single point of failure.

Fair: There is a single point of failure but joining and leaving nodes do not disrupt the search process.

Good: Leaving or failing nodes are quickly detected and do not hamper the search process. Also there is no single point of failure.

Some of the main P2P applications can be graded as follows:

- eDonkey2000: According to [32] this protocol has no messages to signal a leaving node. Shutdown of the central server isn’t supported by the protocol either and would stop any node from initiating a query. Rating: *Bad*;

- Bittorrent: A Bittorrent client only requires a tracker to connect to the swarm of nodes. However failing of a tracker would prevent new nodes from connecting to the swarm. Rating: *Fair*;
- Ad hoc self supervising networks: Due to its distributed and unstructured nature failing or leaving nodes cause no problems in this protocol. Rating: *Good*;
- Unstructured adaptive lookup: In case of a node failing or leaving the network this protocol automatically recalculates the best route to the data [8]. Rating: *Good*;
- Skype p2p internet telephony protocol: Since the Skype protocol is proprietary there is no information on how the protocol handles leaving or failing (super) nodes although a leaving regular node should have no consequence. Rating: N/A (Not known);
- FastTrack protocol: Once connected, nodes cache a list of super nodes. If one fails they can simply connect to another one. Failing of the central server that keeps track of super nodes would prevent new nodes from joining. Rating: *Fair*;
- Session Initiated protocol using DHT: Leaving nodes or super nodes have to unregister with the network. If a (super) node fails this is detected and in case of a super node its leaf nodes will be redistributed among the other super nodes [33]. Rating: *Good*;
- Selective publishing: According to [17] this protocol handles network churn, the joining and leaving of nodes, correctly. This network is also distributed. Rating: *Good*;
- P2PSIP: Depending on the used DHT protocol P2PSIP can handle network churn correctly. When using Chord this is the case [12]. Rating: *Good*;

5.4 Latency

Results found in literature can't be compared since they were measured on different networks and different hardware. They can however provide some indication of the search method's efficiency.

Some of the main P2P applications can be graded as follows:

- eDonkey2000: Latency of queries in this network are highly dependable on the capacity of the server and server load as well as well as latency between client and server. Rating: N/A (Not known);
- Bittorrent: Since queries are handled outside the network latency depends on the indexing website and tracker. Rating: N/A (Not known);
- Ad hoc self supervising networks: No measurements are available. Rating: N/A (Not known);
- Unstructured adaptive lookup: By definition this method has a lower search latency than other query flooding techniques due to the selection of fast nodes to forward the query to. However no measurements are available. Rating: N/A (Not known);
- Skype p2p internet telephony protocol: According to [9] user lookup takes about 3-4 seconds. Rating: N/A (Not known);
- FastTrack protocol: Latency depends on individual nodes, network load and the TTL value. Rating: N/A
- Session Initiated protocol using DHT: It is expected that on a network with 10,000 users, call setup will take one or two seconds with good network conditions [9]. Rating: N/A (Not known);
- Selective publishing: In the test setup a DHT query was started if there were no results using query flooding after 30 seconds. The DHT query would return results in 10-12 seconds [17]. Rating: N/A (Not known);
- P2PSIP: Latency is affected for the most part by the distances between users and when used call setup delay in a P2PsIP based telephony network could range from under 100ms to more than 10 seconds [31]. Rating: N/A (Not known);

Due to missing data on most of the search methods and the incomparable network structures no conclusions can be drawn from these results. Therefore no rating classifications have been defined for this requirement. What can be said is that structured networks are on overall considered to be more efficient in returning search results than super node networks. [34] Furthermore latency in networks with a central index is lower than on those with distributed indexes [33].

5.5 Security support

The grades Bad, Fair, Good can be interpreted for this criterion as follows:

Bad: No security support.

Fair: Traffic on the network relating to search is encrypted.

Good: The solution provides support for all or most of the following security services: DoS prevention, node anonymity, traffic encryption, file authenticity (the file content is what the title claims it is), message authenticity, see e.g., [35].

Some of the main P2P applications can be graded as follows:

- eDonkey2000: All traffic on this network is transmitted unencrypted. No other security measures are available. Rating: *Bad;*
- Bittorrent: Depends on the web site. Implementation of encrypted searching is possible. Authenticity of content can't be guaranteed. Rating: *Fair*;
- Ad hoc self supervising networks: [7] provides no details so the assumption is made none are available. Rating: *Bad*;
- Unstructured adaptive lookup: [8] provides no details so the assumption is made that none of the security services are supported. Rating: *Bad*;
- Skype p2p internet telephony protocol: All Skype communication is encrypted [9]. Content authenticity is not applicable for p2p telephony. DoS attacks are unlikely using the Skype client but might be possible with a client build for that purpose. Rating: *Fair*;
- FastTrack protocol: Due to the nature of the query flooding mechanism, this network is vulnerable to DoS attacks. The protocol also uses uuhash which only uses the hash of parts of the file making it possible for malicious users to corrupt other parts of the file [36]. File authenticity cannot be ensured either. However all traffic on the FastTrack network is encrypted including queries. Rating: *Fair*;
- Session Initiated protocol using DHT: The design in [15] leaves security as an open issue. Rating: *Bad*;

- Selective publishing: [17] provides no details so the assumption is made none are available. Rating: *Bad*;
- P2PSIP: [20] states that P2PSIP is vulnerable to DoS attacks. However P2PSIP does provide encryption support. Rating: *Fair*;

5.6 Use of standardized solutions

The grades Bad, Fair, Good can be interpreted for this criterion as follows:

Bad: The protocol is proprietary and the messaging within the protocol is unknown.

Fair: The protocol does not use a standardized solution but information on the protocol messages is available.

Good: The protocol uses a standardized solution.

Some of the main P2P applications can be graded as follows:

- eDonkey2000: Not based on any standardized solution but since this protocol is unencrypted the messaging is available [32]. Rating: *Fair*;
- Bittorrent: Not based on any standardized solution but available. Rating: *Fair*;
- Ad hoc self supervising networks: Not based on any standardized solution but available. Rating: *Fair*;
- Unstructured adaptive lookup: Not based on any standardized solution but available. Rating: *Fair*;
- Skype p2p internet telephony protocol: Unknown because the inner workings of Skype remain unpublished. Assuming they are not following standards. Rating: *Bad*;
- FastTrack protocol: Unknown because the inner workings of the FastTrack protocol remain unpublished. Assuming they are not following standards. Rating: *Bad*;
- Session Initiated protocol using DHT: SIP is an IETF standard. This solution however uses chord as well Rating: *Fair*;
- Selective publishing: Not based on any standardized solution but available. Rating: *Fair*;
- P2PSIP: P2PSIP is a IETF standard in development. Rating: *Good*;

5.7 Mobility

The grades *Bad*, *Fair*, *Good* can be interpreted for this criterion as follows:

Bad: Not compatible with mobile devices.

Fair: This protocol can be used on mobile devices but the search performance is decreased.

Good: This protocol can be used on mobile devices without a affecting search performance.

Some of the main P2P applications can be graded as follows:

- eDonkey2000: [37] suggests that the eDonkey protocol can be used in combination with the GPRS network. However search latency is increased and switching of access points causes delays as well. Rating: *Fair*;
- Bittorrent: In 2007 the first Bittorrent client for mobile phones was introduced [38]. It offers the same functionality as regular clients but is slower. Rating: *Fair*;
- Ad hoc self supervising networks: Not discussed in [7]. Rating: *Bad*;
- Unstructured adaptive lookup: Not discussed in [8]. Rating: *Bad*;
- Skype p2p internet telephony protocol: The Skype client is available for mobile devices however there is no data on search performance assuming a decrease in performance. Rating: *Fair*;
- FastTrack protocol: No known implementations. Rating: *Bad*;
- Session Initiated protocol using DHT: Not discussed in [15]. Rating: *Bad*;
- Selective publishing: Not discussed in [17]. Rating: *Bad*;
- P2PSIP: This protocol has support for mobile devices but search performance is decreased [39]. Rating: *Fair*;

6. CONCLUSIONS AND FUTURE WORK

In this paper the search requirements of p2p file sharing and p2p telephony were explored as well as several solutions for each p2p search method class. After this the solutions were tested in their ability to satisfy the requirements. An overview of the results of this comparison can be found in Table 2 in Appendix A. The results that differ significantly from their importance rating (e.g. importance high, result bad) have been indicated with a grey background. The green background means the rating matches or surpasses the importance of the requirement. This leads to the conclusion that the P2PSIP protocol and thus the Distributed Has Table search method class is most suited for p2p telephony. For search within file sharing Bittorrent and thus a form central index search class seems to have the most benefits.

Future work could include an attempt to adapt good properties of other search methods within the same search method class for use in the best search methods. More protocols could also be examined and their use for more applications than search within file sharing and telephony.

ACKNOWLEDGEMENTS

I would like to thank Georgios Karagiannis for the review work and guidance he provided. I would also like to thank the other reviewers for their perspective and comments.

REFERENCES

[1] G. Sakaryan, M. Wulff, and H. Unger, "Search Methods in P2P Networks: A Survey," *Innovative Internet Community Systems*, pp. 59-68, 2006.

[2] J. Risson, and T. Moors, “Survey of research towards robust peer-to-peer networks: Search methods,” *Computer Networks*, vol. 50, no. 17, pp. 3485-3521, 2006.

[3] D. Kundur, L. Zhu, M. Merabti *et al.*, "Advances in Peer-to-Peer Content Search." pp. 404-407.

[4] J. F. Kurose, and K. W. Ross, *Computer Networking: A Top-Down Approach*: Addison-Wesley, 2008.

[5] E. Meshkova, J. Riihijärvi, M. Petrova *et al.*, “A survey on resource discovery mechanisms, peer-to-peer and service discovery frameworks,” *Computer Networks*, vol. 52, no. 11, pp. 2097-2128, 2008.

[6] P. Burstein, and A. J. Smith, "Efficient search in file-sharing networks." pp. 1-9.

[7] F. C. Brian, and G.-M. Hector, “Ad Hoc, self-supervising peer-to-peer search networks,” *ACM Trans. Inf. Syst.*, vol. 23, no. 2, pp. 169-200, 2005.

[8] K. Haribabu, D. Reddy, C. Hota *et al.*, "Adaptive lookup for unstructured peer-to-peer overlays." pp. 776-782.

[9] S. A. Baset, and H. G. Schulzrinne, "An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol." pp. 1-11.

[10] J. Liang, R. Kumar, and K. W. Ross, "The FastTrack overlay: A measurement study," *Computer Networks*, vol. 50, no. 6, pp. 842-858, 2006.

[11] B. Hari, M. F. Kaashoek, K. David *et al.*, "Looking up data in P2P systems," *Commun. ACM*, vol. 46, no. 2, pp. 43-48, 2003.

[12] I. Stoica, R. Morris, D. Liben-Nowell *et al.*, "Chord: a scalable peer-to-peer lookup protocol for internet applications," *IEEE/ACM Trans. Netw.*, vol. 11, no. 1, pp. 17-32, 2003.

[13] S. Ratnasamy, P. Francis, M. Handley *et al.*, "A scalable content-addressable network." pp. 161-172.

[14] A. Rowstron, and P. Druschel, "Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems," *Middleware 2001 : IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg, Germany, November 12-16, 2001. Proceedings*, p. 329, 2001.

[15] S. Kundan, and S. Henning, "Peer-to-peer internet telephony using SIP," in Proceedings of the international workshop on Network and operating systems support for digital audio and video, Stevenson, Washington, USA, 2005.

[16] IETF, "SIP: Session Initiation Protocol," rfc3261, 2002.

[17] B. T. Loo, R. Huebsch, I. Stoica *et al.*, "The Case for a Hybrid P2P Search Infrastructure," *Peer-to-Peer Systems III*, pp. 141-150, 2005.

[18] "Peer-to-Peer Session Initiation Protocol (p2psip), Charter," 05-26, 2009; http://www.ietf.org/html.charters/p2psip-charter.html.

[19] D. A. Bryan, B. B. Lowekamp, and M. Zangrilli, "The design of a versatile, secure P2PSIP communications architecture for the public internet." pp. 1-8.

[20] J. Seedorf, F. Ruwolt, M. Stiemerling *et al.*, "Evaluating P2PSIP under Attack: An Emulative Study." pp. 1-6.

[21] "LionShare," may 2, 2009; http://lionshare.its.psu.edu/.

[22] M. D. Bontrager, "Peering into the future: Peer-to-Peer technology as a model for distributed joint battlespace intelligence dissemination and operational tasking," School of advanced airpower studies, Air University Alabama, 2001.

[23] K. Tomoya, and Y. Shigeki, "Application of P2P (peer-to-peer) technology to marketing." pp. 372-379.

[24] A. Nafaa, S. Murphy, and L. Murphy, "Analysis of a large-scale VOD architecture for broadband operators: A P2P-based solution," *IEEE Communications Magazine*, vol. 46, no. 12, pp. 47-55, 2008.

[25] J. Mercier. "Skype Numerology: 15 Milion today," 25-4, 2009; http://skypenumerology.blogspot.com/2009/01/15-million-today.html.

[26] *eBay Inc. Reports First Quarter 2009 Results*, eBay Inc., San Jose, 2009.

[27] J. Pouwelse, P. Garbacki, D. Epema *et al.*, "The Bittorrent P2P File-Sharing System: Measurements and Analysis," *Peer-to-Peer Systems IV*, pp. 205-216, 2005.

[28] N. Daswani, H. Garcia-Molina, and B. Yang, "Open problems in data-sharing Peer-to-Peer systems," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2003, pp. 1-15.

[29] C. M. Huang, T. H. Hsu, and M. F. Hsu, "Network-aware P2P file sharing over the wireless mobile networks," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 1, pp. 204-210, 2007.

[30] G. Caizzone, A. Corghi, P. Giacomazzi *et al.*, "Analysis of the Scalability of the Overlay Skype System." pp. 5652-5658.

[31] Z. Chunhong, S. Juwei, L. Lichun *et al.*, "Signaling Latency Analysis of Peer-to-Peer SIP Systems." pp. 505-509.

[32] O. Heckmann, and A. Bock, *The eDonkey 2000 Protocol*, 08/2002, Darmstadt University of Technology, Darmstadt, 2002.

[33] M. Weltz, *Peer-to-Peer audio conferencing using SIP: P2P Architecture*, Stockholm, 2007.

[34] Y. Yang, R. Dunlap, M. Rexroad *et al.*, "Performance of full text search in structured and unstructured peer-to-peer systems."

[35] D. S. Wallach, "A Survey of Peer-to-Peer Security Issues," *LNCS series*, vol. 2609/2003, pp. 253-258, 2003.

[36] F. Benevenuto, C. Costa, M. Vasconcelos *et al.*, "Impact of peer incentives on the dissemination of polluted content," in Proceedings of the 2006 ACM symposium on Applied computing, Dijon, France, 2006.

[37] T. Hoßfeld, K. Tutschku, and F. U. Andersen, "Mapping of file-sharing onto mobile environments: Feasibility and performance of edonkey with GPRS." pp. 2453-2458.

[38] I. Kelényi, G. Csúcs, B. Forstner *et al.*, "Peer-to-Peer File Sharing for Mobile Devices," *Mobile Phone Programming*, pp. 311-324, 2007.

[39] M. Matuszewski, and E. Kokkonen, "Mobile P2PSIP - Peer-to-Peer SIP Communication in Mobile Communities." pp. 1159-1165.

APPENDIX A: COMPARISON RESULTS

	Central Index		Query Flooding		Hierarchical Overlay		Distributed Hash Table		
Requirement	eDonkey 2000	Bittorrent	Ad hoc self supervising networks	Unstructured adaptive lookup	Skype p2p internet telephony protocol	FastTrack protocol	Session Initiated protocol using DHT	Selective publishing	P2PSIP
Scalability	Bad	Fair	Fair	Fair	Good	Fair	Good	Fair	Good
Completeness of results	Good	Good	Fair	Fair	Good	Fair	Good	Fair	Good
Dependability	Bad	Fair	Good	Good	N/A	Fair	Good	Good	Good
Latency	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Security support	Bad	Fair	Bad	Bad	Fair	Fair	Bad	Bad	Fair
Use of standardized solutions	Fair	Fair	Fair	Fair	Bad	Bad	Fair	Fair	Good
Mobility	Fair	Fair	Bad	Bad	Fair	Bad	Bad	Bad	Fair

Table 2: Results of the requirement and solution comparison