Computer Science and Information Systems 00(0):0000–0000 https://doi.org/10.2298/CSIS123456789X

Intelligent query processing in P2P networks: semantic issues and routing algorithms

AL Nicolini[1], CM Lorenzetti[1], AG Maguitman[1], and CI Chesñevar[1]

Institute for Computer Science and Engineering (ICIC)
Universidad Nacional del Sur - CONICET, Bahía Blanca, Argentina
{aln, cml, agm, cic}@cs.uns.edu.ar

Abstract. P2P networks have become a commonly used way of disseminating content on the Internet. In this context, constructing efficient and distributed P2P routing algorithms for complex environments that include a huge number of distributed nodes with different computing and network capabilities is a major challenge. In the last years, query routing algorithms have evolved by taking into account different features (provenance, nodes' history, topic similarity, etc.). Such features are usually stored in auxiliary data structures (tables, matrices, etc.), which provide an extra knowledge engineering layer on top of the network, resulting in an added *semantic* value for specifying algorithms for efficient query routing. This article examines the main existing algorithms for query routing in unstructured P2P networks in which *semantic aspects* play a major role. A general comparative analysis is included, associated with a taxonomy of P2P networks based on their degree of decentralization and the different approaches adopted to exploit the available semantic aspects.

1. Introduction

A peer-to-peer network (or just P2P network) is a computing model present in almost every device, from smartphones to large-scale servers, as a way to leverage large amounts of computing power, storage, and connectivity around the world. In a P2P network, each peer can act indistinctly as a client and a server and can collaborate in order to share information in a distributed environment without any centralized coordination. These systems are vulnerable to security problems, abuse, and other threats, and consequently, it is necessary to be resilient to different forms of attacks, to have mechanisms to detect and remove poisoned data [20,72], and to distinguish spammers from honest peers [92,122,94]. Despite these issues, P2P networks are widely used for large-scale data sharing, content distribution, and application-level multicast working with a tolerable waiting time for the users [86,97,144].

P2P technologies have demonstrated great potential to support distributed information retrieval. The typical information retrieval problem in P2P networks involves finding a set of documents in the network that are relevant to a given query. To better describe the problem of information retrieval in P2P networks, it is useful to identify a number of salient features, as illustrated in Figure 1. In a P2P information retrieval network each device or node (peer) maintains a collection of documents available to share with other peers. In order for those peers to interact with each other, several components are required to support search among peers associated with a given query. These components include a routing algorithm, routing tables, indices, and an established protocol that manages the queries that each node can handle [56,48,121]. The routing algorithm determines how a node searches for information by sending query messages to other peers. When a peer receives a query message, it attempts to retrieve relevant documents from its own collection, forwarding as well the query to other peers in the network.

In the context of P2P networks, it is necessary to distinguish between the *physical* network and the *logical* network. The former consists of real physical connections between devices while the latter is a topology that emerges from the peers' interaction. Since the interaction between peers can be guided by their semantic relations, *semantic communities* commonly emerge in logical networks [36,22].

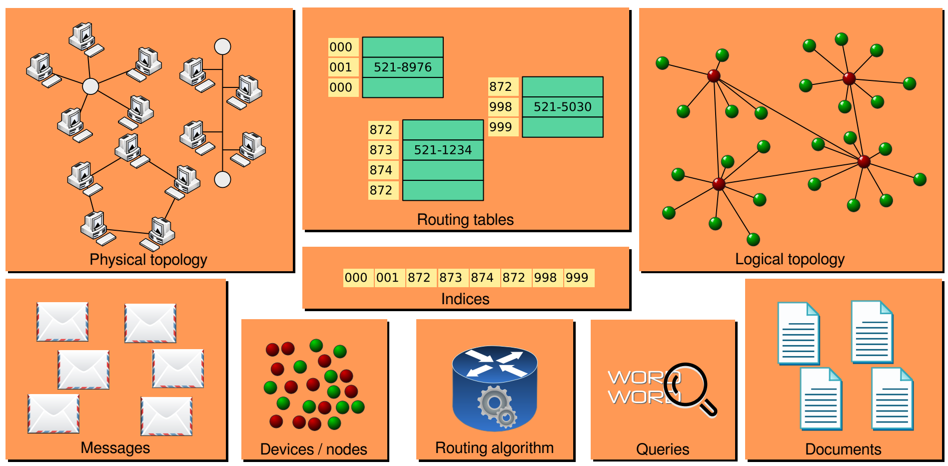

Fig. 1. Conceptual elements which characterize a P2P network.

Different methodologies and techniques for information retrieval on P2P networks have been proposed [129], providing alternative approaches to exploit the concept of social communities on the Internet. Some of the benefits which result from applying P2P information retrieval networks are the following:

- By their very nature, P2P networks do not need a centralized administration, being self-organizing and adaptive (in the sense that peers can enter and leave the network without any external control).
- Peers can have access to several storage and processing resources available from different computers and devices in the network.
- Since pure P2P networks are distributed and decentralized, they tend to be fault-tolerant and with a good load balance for handling network traffic.

Over the years, the Internet has become increasingly restricted to client-server applications. Unfriendly protocols and firewalls are examples of aspects that restrict and limit the use of Internet. To some extent, P2P technologies can be thought of as a way of returning the Internet to its original cooperative design, in which every participant creates as well as consumes [101]. The emerging P2P networks could thus empower almost any group of people with shared interests such as culture, politics, health, etc.

In this article, we present a review of literature solutions to the information retrieval problem in unstructured P2P networks and a description of the main techniques for routing queries in structured and semi-structured P2P systems. Our analysis includes a novel classification of these systems, putting special emphasis on their semantic aspects and the different existing routing algorithms. While existing algorithms have facilitated the implementation of robust distributed architectures, there are still several limitations faced by current search mechanisms. Indeed, the information retrieval problem is more complex than the traditional problem of searching for resources based on object identifiers or names. Over the years, the information retrieval community has developed numerous document retrieval techniques for centralized search. However, these methods cannot be directly applied to P2P information retrieval networks, where search is not centralized since documents are distributed among a large number of repositories. Given the information explosion that we have experienced in the last years, such new capabilities are an important step for making P2P networks effective in many applications that go beyond simple data storing. There has been previous research work which provided the background for our analysis: [147] presents a review of some early methods developed to address information retrieval problems in P2P networks. An empirical comparison of some of these methods is presented in [132] and a more recent survey of the major challenges for P2P information retrieval is presented in [129].

The remainder of the article is organized as follows: Section 2 presents some background concepts used in the rest of the paper, including a description of the main components of P2P networks and a classification of P2P search algorithms. Section 3 presents a summary of the most important search algorithms in P2P networks, highlighting those that make use of semantic aspects. Section 4 provides a comparison and classification of the algorithms presented in Section 3. Finally, the conclusions derived from this analysis are presented in Section 5.

2. Background

According to [117], a P2P network is, in its pure form, “a distributed system in which every peer communicates with other peers without the intervention of centralized hosts”. In real-world P2P networks, the participating peers are typically computers to be found at the edge of the network, in people’s offices or homes [77], Thus, a P2P network turns out to be formed by a set of machines, which offer a wide range of capabilities when considering storage and Internet access speeds, being attractive for different computing tasks (such as file sharing, media streaming, and distributed search). P2P networks have usually no centralized directory, being *self-organizing*, with the ability to adapt to different circumstances associated with the participating peers (e.g. joining in, failing or departing from the network). It is worth noticing that the use of a common language ensures that the communication between peers is *symmetric* for both the provision of services and communication capabilities. From this symmetry the P2P network can also be characterized as *self-scaling*, since each peer that joins the network adds a new computational resource to the available total capacity [32,112,29]. There are many important challenges specific to P2P networks [45], such as how to administrate resources properly, how to provide an acceptable quality of service while guaranteeing robustness and availability of data, etc. Reviews of different P2P frameworks and their applications can be found in [24,105,89].

The rest of this section will present different dimensions relevant for assessing P2P networks. First, in subsection 2.1 we present a common classification for P2P networks. Then, subsection 2.2 introduces the concept of *semantic aspect* in the context of P2P information retrieval. Subsection 2.3 introduce a novel taxonomy for routing algorithms based on semantic aspects. Subsection 2.4 present the importance of distributed hash tables for the implementation of routing algorithms mainly for structured topologies. To conclude the section, subsection 2.5 introduces some of the salient applications of the P2P technologies.

2.1. Network classification

A common approach to classify P2P networks is based on their *degree of centralization*, which results in three possible alternatives:

- **Centralized**: These P2P networks have a monolithic architecture with a single server that allows transactions between nodes and keeps track of where content is stored. For example, *Napster* [100] had a constantly-updated directory hosted in a central location (the Napster website). This system was extremely successful before its legal issues [50]. Clearly, this centralized approach scales poorly and has a single point of failure.
- **Decentralized**: These are systems where there is no centralized directory, since each peer acts as a client and a server at the same time. *Gnutella* [111] is an example of this architecture, where the network is formed by nodes that join and leave the system. Several factors motivate the adoption of decentralized networks such as privacy control, availability, scalability, security, and reliability [107]. On the downside, to find a file in a decentralized network a node must query its neighbors. In its basic form, this method is called *Flooding* and is extremely unscalable, generating large loads on the network participants.
- **Hybrid**: These systems have no central directory server and therefore can be seen as decentralized networks. However, they have some special peers or super-peers with extra capabilities. *FreeNet* [41,4] is an example of such system and there is a growing interest on this kind of P2P architectures, which supports a hash-table-like interface [110,124,114,149,63].

The network topology is another common classification criterion for P2P networks, allowing to identify two distinctive groups:

- **Structured**: In structured P2P networks the nodes are organized into a specific topology. This organization ensures that any node can search the network for a resource, providing as well a good response time. The most common type of structured P2P network is based on a distributed hash table (DHT) that provides a lookup service similar to a hash table: (key, value) pairs are stored in a DHT, and any participating node can efficiently retrieve the value associated with a given key. This approach is adopted by *Chord* [124], *Pastry* [114] and *Tapestry* [149], among others.
- **Unstructured**: In unstructured P2P networks no structure is pre-established over the network, but rather these networks are formed by nodes that randomly build connections to each other. Because of their lack of structure, unstructured networks are easy

to build and highly robust. However, a major limitation of these networks is their poor effectiveness in finding resources. The simplest algorithm used on unstructured P2P networks is based on propagating the query message through all the network, leading to a high amount of traffic. Additionally, it is not possible to ensure that search queries will be eventually resolved. Some examples of this type of systems are *Gnutella* [111] and *KaZaa* [1].

Figure 2 shows a diagram with the classification of P2P systems, identifying the relationships between the different degrees of centralization and the possible topology of the network. Structured topologies are frequently related to centralized or hybrid systems in order to take full advantage of both features. However, unstructured topologies have random connections and in general any peer is equivalent to the others, so that they are strongly associated with the concept of decentralization. The items marked with a star correspond to the central topics on which we will focus on the rest of this survey.

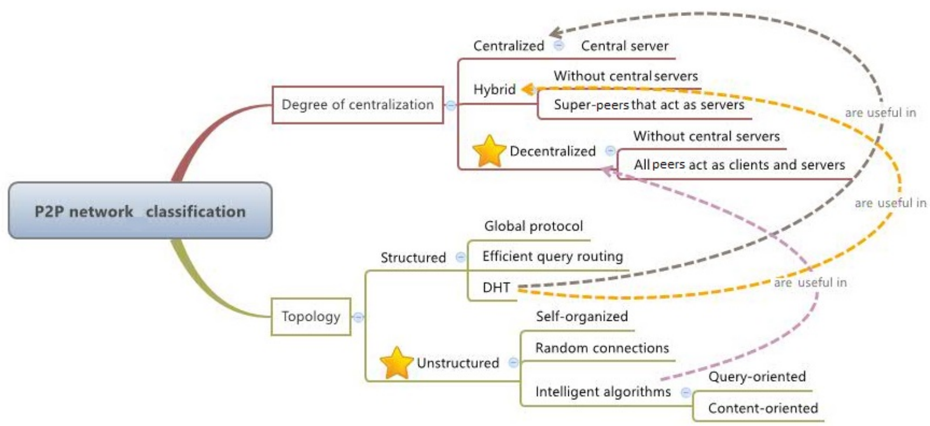

Fig. 2. Classification of P2P networks in terms of their degree of centralization and associated topology.

Search methods depend on the underlying network structure. As discussed before, in the case of pure unstructured networks there is no specific pattern for the organization of its nodes, resulting in a random topology. As a consequence, these networks have relatively low search efficiency when contrasted with structured and hybrid ones, as a single query can generate massive amounts of traffic even if the network has a moderate size [112]. Thus, many alternatives have been studied to improve the basic flooding approach in unstructured networks, such as random walk [88,71,145,28,80], directing the search towards potentially useful nodes [21,147,138,120], or clustering peers by content [44,133,33] or interest [123,93,108].

2.2. Semantic Aspects

A possible solution to improve communication overhead and scalability in large-scale unstructured P2P systems is to forward queries to a group of peers that are known to be potentially useful to answer the query. The selection of potentially useful peers is typically based on the peers' past activity or their semantic similarity to the original query [43,123,135,74,85,78,130]. Identifying peers based on their potential to answer a query in a useful way requires associating semantic aspects with peers.

A *semantic aspect*, in the context of a P2P network, is a feature or a set of features that allows recognizing the semantic of the data stored in a node. Semantic aspects can be exploited by routing algorithms to help peers predict which other peers have knowledge useful to respond to a query. Topical information, past experience, and node-state information are examples of such semantic aspects.

In algorithms that use topical information to compute topic similarity, each peer stores profiles of other peers. A neighbor profile is information that a peer maintains to describe the content stored by a neighbor. By analyzing neighbor profiles, peers try to increase the probability of choosing appropriate peers to route queries. Algorithms that use topic similarity to guide the search process in a P2P network lead to the spontaneous formation of semantic communities through local peer interactions [22].

A key concept associated with the use of topic similarity to guide the search process is "semantic locality" [140]. Traditionally, the notion of semantic locality has been used to refer to the ability to store information about peers offering semantically close services. This ability can be used to index and locate content, complementing the current service discovery mechanisms in a Grid and in the Web [115,150,116,79]. Semantic locality has also been defined as "a logical semantic categorization of a group of peers sharing common data" [119]. With the help of locality information, an unstructured P2P network allows to design more informed mechanisms for routing queries, mitigating the complexity of the search process [102,103].

Another alternative for capturing semantic aspects consists in storing past experiences from the interaction of a peer with its neighbors. This approach does not require storing nodes' profiles. Instead, a peer keeps track of valuable routing information (such as the number of hits per node or peer availability) and uses this information to select the most active peers to forward a query [49]. The main problem with this approach is that it strongly benefits those peers that store large amounts of data, penalizing less resourceful peers that may also offer relevant material for specific topics.

A number of algorithms use heuristic information based on the state of the nodes and their past performance to select candidate nodes. There are many heuristics that can be considered; some of them are based on the analysis of the latency or the response time of specific nodes [151]. Clearly, in such cases, additional specific data must be collected and stored for computing the associated heuristic. For example, in [82] a heuristic-based query routing algorithm is presented. This algorithm collects a plurality of metrics for each host that it is aware of in a P2P network, most often by host information or query hit messages.

The metrics collected aid in determining a set of P2P hosts best able to fulfill a query message, without having knowledge of specific content. The metrics collected also aid in managing query messages in that they determine when to drop query messages or when to resend query messages. The choice of the heuristic is a very important step in the process of developing an algorithm. In [141] a study of the *DirectedBFS* algorithm implemented under different heuristics is presented. In this algorithm, in order to intelligently select neighbors, a node maintains simple statistics on its neighbors, such as the number of results received through that neighbor for past queries, or the latency of the connection with that neighbor. From these statistics, the authors develop a number of heuristics to select the best neighbor to send the query, such as:

– Select the neighbor that has returned the highest number of results for previous queries.
– Select the neighbor that returns response messages that have taken the lowest average number of hops. A low hop-count may suggest that this neighbor is particularly close to nodes containing useful data.
– Select the neighbor that has forwarded the largest number of messages (all types). A high message count would imply that this neighbor is stable and it can handle a large flow of messages.
– Select the neighbor with the shortest message queue. A long message queue implies that the neighbor's pipe is saturated, or that the neighbor has died.

In summary, the use of semantic aspects helps to select the most promising nodes to route queries with the purpose of implementing more informed search strategies.

2.3. Algorithm classification based on semantic aspects

In unstructured P2P networks, routing algorithms can be classified into *content-oriented* or *query-oriented*, based on the semantic aspects and the decision-making criteria used to route a query [26].

– **Content-oriented**: these routing algorithms use metadata extracted from the shared content of each peer to build a local index with global information. This index provides each peer with an approximate view of the network content and other peers' profiles. Hence, peers will be able to route efficiently their queries, improving the retrieval effectiveness. Nodes' profiles are the most used semantic aspects in content-oriented routing algorithms [43,76].
– **Query-oriented**: these routing algorithms exploit the historical information of past queries and query hits to route future queries. Past experience based on query hits is the most used semantic aspect in query-oriented routing algorithms [131,81].

Content-oriented algorithms produce a very large number of messages to build their associated indices. In contrast, query-oriented methods are more advantageous, as no excessive network overhead is required for building the routing indices. Recently, efficient approaches to content-oriented routing algorithms have been proposed in the context of content-centric networking [39,146]. In content-centric networking, a data object is retrieved based on its content identity instead of the IP address of the node on which it resides.

2.4. Distributed Hash Tables in P2P systems

Distributed Hash Tables (DHTs) are data structures for indexing data using a distributed approach. DHTs provide a powerful tool that has changed the way resources and information are shared. In structured P2P systems, the data objects are stored by a globally-agreed scheme. In this context DHTs have turned out to be one of the most highly used approaches [52]. In P2P systems implemented with DHTs, each peer represents a hash table bucket with a global hash function. Search in this kind of systems is guaranteed and efficient since it typically involves logarithmic time with respect to the overlay network size. A popular P2P system based on DHTs is Kademlia [91], which includes several desirable characteristics that were not present in previous DHT-based approaches. Kademlia minimizes the number of configuration messages that every node needs to send in order to learn about each other. In this system, each node has enough knowledge to be able to proceed with query routing using low-latency paths. Recent work on Kademlia has been oriented towards analyzing the resilience against failing nodes and communication channels [61] and secure and trustable distribution aggregation [57]. Viceroy [90] is another P2P system whose relevance lies on being the first P2P system to combine a constant degree with a logarithmic diameter, while still preserving fairness and minimizing congestion. This is achieved through a quite complex architecture that guarantees with high probability that the congestion of the network is within a logarithmic factor of the optimum. Later work on Viceroy resulted in Georoy [54], an algorithm for efficient retrieval of information based on the Viceroy P2P algorithm. Unlike Viceroy, Georoy establishes a direct mapping between the identification of a resource and the node which maintains information about its location. In spite of all their advantages, it must be remarked that systems based on DHTs suffer from limitations in terms of robustness and search flexibility. A good P2P structured system needs cooperation among peers to maintain flexibility and credibility. This assumption is particularly strong, as not all devices on the Internet connected through the network are necessarily stable and reliable.

2.5. Applications of P2P systems

Many software applications that gained popularity among a large community of users, such as *eMule* [14] and *PopCorn Time* [17], operate on P2P networks. In both cases, these applications use P2P technologies to stream audio and video to their end users, generating a considerable portion of the overall traffic on the Internet and requiring a large amount of energy consumption [34].

Regarding to educational settings, P2P technologies have allowed institutions to share files globally, as is the case of the *LionShare* project[96]. Another popular distributed application is *Bitcoin* and its alternatives, such as *Peercoin* and *Nxt*, which are P2P-based digital cryptocurrencies. *Bitcoin* [51] is a P2P system where transactions take place directly among users, without an intermediary. Network nodes are in charge of verifying these transactions, which are eventually recorded in a public distributed ledger (called the blockchain). *Bitcoin* has no central repository (nor administrator) and is known as the first decentralized digital currency [98].

Another area where P2P technologies are becoming increasingly important is social networking. Currently, the social web is mostly dominated by centralized social networks such as *Twitter*, *Facebook* and *MySpace* [18,15,16] and as a consequence it is limited by the centralization in the use of the information and the potential loss of control of the privacy of the information by the users. These social networks create the illusion that users are directly connected to each other. However, the centralized servers belonging to companies are the ones in charge of controlling the data and the interactions among users. The lack of control of the users on their data is a problem that is aggravated by the terms of services, which are often unclear or have notable disadvantages, and by the occasional changes in the privacy policy. Distributed technologies offer a possible solution to the problem of centralization. With this approach, servers (peers) can communicate with each other without the intervention of a central server. This schema allows that data, and the control over them, to be distributed among all users and their servers. P2P networks offer a way to relax the constraints of centralized control, resulting in systems that are decentralized, concurrent and with collective or emerging behaviors. These features make P2P an attractive technology to support social networks. An example of open source distributed social network is *BuddyCloud* [2], which allows software developers to share their applications, supplemented with chats and videos. Another distributed social network is *Diaspora** [3], whose policy allows the decentralization in the use of information. In this social network profiles are stored in users' personal web servers, allowing them to have full control of the content they share and to have absolute knowledge of where the content is stored and who has access to it. Other examples of distributed social networks are *Friendica* [5], *GNU social* [6], *Mastodon* [8], *Minds* [9], *Kune* [7] and *Twister* [10].

Clustering is an important data mining issue, especially for large and distributed data analysis. Distributed computing environments such as P2P networks involve separated sources, distributed among the peers. According to unpredictable growth and dynamic nature of P2P networks, data of peers are constantly changing. Due to the high utilization of computing and communication resources and privacy concerns, processing of these types of data should be applied in a distributed way and without central management. In this scenario, clustering algorithms became important to organize the peers among the network. An example of this kind of algorithm is *GBDC-P2P* [27]. The *GBDC-P2P* algorithm is suitable for data clustering in unstructured P2P networks and it adapts to the dynamic conditions of these networks. In the *GBDC-P2P* algorithm, peers perform data clustering with a distributed approach only through communications with their neighbors.

Distributed data storage is another area in which P2P technologies have proven to be helpful. Distributed databases allow quick access to data stored throughout a network, and have different capabilities (e.g. some provide rich query abilities whereas others are restricted to a key-value store semantics). Google's Bigtable [12], Amazon's Dynamo [13], Windows Azure Storage [19], and Apache Cassandra [11] (formerly Facebook's data store [59]) are examples of distributed databases. In P2P network data storage, the user can usually reciprocate and allow other users to use their computer as a storage node as well. Information may or may not be accessible to other users depending on the design of the network.

3. Routing algorithms

In this section, we analyze around forty different algorithms as well as some of their variants related to intelligent query routing in P2P networks. In particular, as discussed previously, we will focus on unstructured P2P networks in which semantic issues play a mayor role. We provide a brief description of each algorithm along with the corresponding references.

3.1. Routing algorithms in structured P2P networks

In a structured P2P system, the topology that defines the connections among peers and data locations is predefined. This pre-established topology is exploited by search mechanisms that take advantage of these pre-defined relations among peers. Even when using a distributed hash table (DHT), structured systems may differ on the data structures used for implementing it (e.g. some of them may rely on flat overlay structures while others might be based on hierarchical overlay structures). A benefit of DHTs is the possibility to exploit the structure of the overlay network for sending a message to all nodes, or a subset of nodes, ensuring a threshold for the overall execution time involved. It is not natural to implement a search algorithm with DHT in unstructured networks due to their lack of structure. However, some authors have explored their application in unstructured and semistructured networks [62].

Some flat data structures (Figure 3) include ring, mesh, hypercube, and special graphs such as the de Bruijn graph [46]. For example, *Chord* [124] uses a ring data structure with node IDs. Each node keeps a table that contains the IP addresses of those nodes that are half of the ID ring away from it. A key k is mapped to a node A whose ID is the biggest that does not exceed k. In the search process, A forwards the query for key k to $succ(k)$ (node in A's table with the highest ID that is not larger than k). In this way, a query can be forwarded until the node that holds the key is reached. The so-called "finger table" speeds up the lookup operation, ensuring an execution time of $O(logN)$.

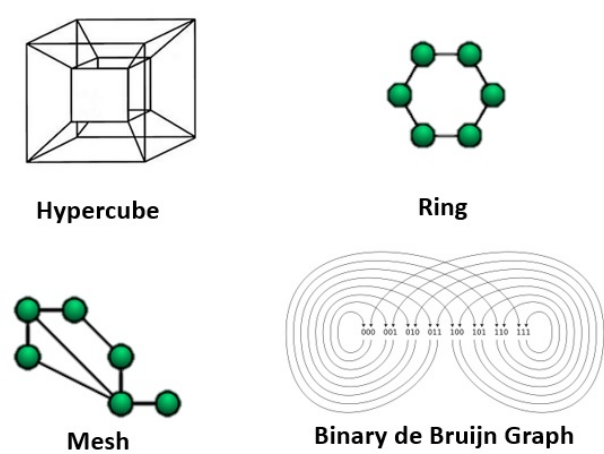

Fig. 3. Some examples of possible structured topologies.

Pastry [114] is based on a tree data structure which can be considered a generalization of a hypercube. Each node A keeps a leaf set L. For every node A, the set L consists of those $L/2$ nodes whose IDs are nearest to and smaller than the ID of A, along with the set of $L/2$ nodes whose IDs are nearest to and bigger than the ID of A. This set ensures the correctness of the process. Every Pastry node also keeps a routing table of references to other nodes of the same ID space. In Pastry, given a search query q associated with a key k, a node A forwards q to a node whose ID is the nearest to k among all the nodes known to A. Node A tries then to find a node in its leaf set. If that node does not exist, A looks for a candidate node in its routing table whose ID shares a longer prefix with k than A. If this node does not exist either, A forwards q to a node whose ID has the same shared prefix than A but is numerically closer to k than A. In this way, each Pastry node ensures an execution time of $O(logN)$. The approach presented in [149] called *Tapestry* is similar to the previous one. They differ in the underlying routing algorithm and in the approach taken to exploit the locality. Tapestry also ensures an execution time of $O(logN)$.

In *CAN* [110] DHTs are implemented using a d-dimensional toroidal space divided into hyper-rectangles, which define different zones. Each of these zones is controlled by a particular node. Keys are mapped with a hash function to points in the d-dimensional space. Each node has a routing table that consists of all of the other nodes that are in its d-dimensional space. A node A is in the same space of another node B if the zone of B shares a $(d-1)$-dimensional hyperplane with the zone of A. Given a query q associated with a key k, a node forwards q to another node according to its routing table whose zone is the nearest to the zone of the node responsible for key k.

The *NetSize-aware* protocol introduced in [148] is based on *CAN*. The main objective of this algorithm is to solve the problem of search flexibility in DHT. This algorithm preserves CAN's simplicity, providing a greedy routing algorithm based on DHT. *NetSize-aware* uses a binary partition tree algorithm to determine the underlying network topology. Simulation results show that this approach is resilient, efficient and improves the performance of *CAN*.

KaZaA [1] is a P2P file-sharing technology that was commonly used to exchange MP3 music files and other file types (such as videos, applications, and documents) over the Internet. Its architecture is based on a two-tier hierarchy in which some nodes are distinguished as *supernodes*. Supernodes are those nodes with the fastest Internet connection and best CPU power. Each supernode is responsible for indexing the files of the nodes that it handles. The use of supernodes with better computing capabilities than regular nodes allows the system to perform better than the local-indices approach respect to lower susceptibility to bottlenecks, and similar resilience to churn (where the churn rate can be defined as a measure of the number of individuals or items moving out of a collective group over a specific period of time). However, this system suffers the problem of the resulting overhead associated with exchanging index information between regular nodes and supernodes [143,84]. In [65] a routing algorithm that is also structured in two layers is proposed: *SkipNet* layer and *Small-World* layer. The first layer routes the queries based on a numerical ID and the second layer routes the queries using a Small-World topology (see [137] for a pioneering study of Small-World networks).

An efficient P2P information retrieval system called *pSearch* is presented in [127]. *pSearch* supports semantic-based full text searches and avoids the scalability problem of certain systems that employ centralized indexing. In *pSearch* documents are organized based on their vector representations generated by information retrieval algorithms based on the vector space model and latent semantic indexing. This organization results in more efficiency and accuracy, as the search space for a particular query is defined on the basis of related documents.

The growth of intercommunication between computers gives systems the chance to operate more efficiently, by better supporting the cooperation between individual components. *AFT* [106] is an overlay that adapts to a changing number of nodes in a P2P network and is resilient to faults. The *AFT* overlay is designed to be a solution for systems that need to share transient information, performing a synchronization between various components, such as in mobile ad-hoc networks, urban networks, and wireless sensor networks. The operations supported by the overlay, such as joining, leaving, unicast transmission, broadcast sharing and maintenance can be accomplished in time complexity of $\mathcal{O}(\sqrt{N})$, where N is the number of nodes which are part of the structure.

In [146] a novel framework is introduced, based on implementing a hybrid forwarding mechanism. This approach allows discovering content in a proactive or reactive way based on content characteristics. The proposed framework classifies time-sensitive data utilizing content identifiability and content name prefixes, aiming at applying the most suitable strategy to each category. For proactive content dissemination they propose a Hierarchical Bloom-Filter based Routing algorithm (see [35] for a detailed review of the concept of bloom filters). A Hierarchical Bloom-Filter is structured in a self-organized geographical hierarchy, which makes the approach scalable to large metropolitan Vehicular Ad-Hoc Networks (VANETs).

An approach presented in [79] characterizes the notion of semantic-based sub-spaces as a basis for organizing the huge search space of large-scale networks. Each sub-space consists of a set of participants that share similar interests, resulting in semantic-based Virtual Organizations (VOs). Thus the search process occurs within VOs where queries can be propagated to the appropriate members. The authors propose a generic ontological model that guides users in determining the desired ontological properties and in choosing the “right” VOs to join. DHTs are used to index and lookup the hierarchical taxonomy in order to implement the ontology directory in a decentralized manner. Even though the ontology-based model facilitates the formation of the VOs, searching and sharing efficiently is still a major challenge due to the dynamic and large-scale properties of the search space. In order to efficiently share and discover resources inside VOs an infrastructure called *OntoSum* is proposed.

Security is an important feature in all type of networks, especially in P2P networks where every participant requests and provides information without any centralized control. To prevent structured overlay networks from being attacked by malicious nodes, a symmetric lookup-based routing algorithm referred to as *Symmetric-Chord* is presented in [87]. This algorithm determines the precision of routing lookups by constructing multiple paths to the destination. The selective routing algorithm is used to acquire information on the neighbors of the root. The authenticity of the root is validated via consistency shown between the information ascertained from the neighbors and information from the yet-to-be-verified root, resulting in greater efficiency of resource lookup. Simulation results demonstrate that *Symmetric-Chord* has the capability of detecting malicious nodes both accurately and efficiently, so as to identify which root holds the correct key, and provides an effective approach to the routing security for the P2P overlay network.

Another approach that implements an attack detection method is presented in [66]. The authors propose a routing table “sanitizing” approach that is independent of a specific attack variant. The proposed method continuously detects and subsequently removes malicious routing information based on distributed quorum decisions, and efficiently forwards malicious information findings to other peers which allows for progressive global sanitizing.

In [23], the authors have proposed a scalable solution for lookup acceleration and optimization based on the de Bruijn graph with right shift. The proposed solution is principally based on the determination and elimination of the common string between source and destination. This procedure is executed locally at the current requestor node. The performance aspects of the proposed model have been validated through simulation results developing a specific Java program. Among other approaches that use de Bruijn graphs we can cite D2B [53], DH-DHT [99] and Koorde [70].

3.2. Routing algorithms in semi-structured or loosely structured P2P networks

In loosely structured P2P networks the overlay structure is not strictly specified. The emerging structure turns out to be formed in a probabilistic way, or defined by some underlying topology. Thus, searching in this kind of networks depends on the overlay structure and how the data is stored [125]. *FreeNet* [41,4] is a P2P loosely structured system designed for protecting the anonymity of data sources. This scheme is based on the DHT interface, where each node has a local data repository and an adaptable routing table. These tables have information about addresses of other nodes and the possible keys stored in these nodes. Searches are performed in the following way: let us assume that node A is the query-issuing node and it generates a query q for a data item with key k. First, A looks up its data repository. If the file is found locally, q is resolved. Otherwise, q is forwarded to the node B whose key is nearest to k according to As routing table. Then, node B performs a similar computation. This procedure continues until the search process terminates. During this process, a node may not forward the query to the nearest-key neighbor because that neighbor is down or a loop is detected. In such cases, this node tries to contact the neighbor with the second nearest key. A TTL (time to live) limit is specified to restrict the number of messages in the query routing process. If the data item is found, the file is returned to the query-issuing node in the reverse path of the query. Each node (except the last one on the query path), creates an entry in the routing table for the key k. To bring anonymity, each node can change the reply message and claim itself or another node as the data source.

PHIRST [113] is a system that aims at facilitating full-text search within P2P databases and simultaneously takes advantage of structured and unstructured approaches. In a similar way to structured approaches, peers publish first terms within their document space. The main difference with respect to other algorithms is that frequent terms can be quickly identified and do not need to be stored exhaustively, thus reducing the storage requirements of the system. In contrast, during query lookup agents use unstructured search to compensate for the lack of fully published terms. In this way the costs of structured and unstructured approaches are balanced, achieving a reduction in the costs involved in the queries that are generated in the system. There are other kinds of semi-structured P2P networks where the network is divided into different subnets, resulting in a topology based on the peers' interests. In [93] a system is presented where nodes are clustered according to their interests [33] to form a P2P overlay network of multilayer interest domains. Three types of nodes are distinguished: active nodes, super-nodes, and normal nodes. Each active node acts as a router providing information that facilitates query routing information at the cluster level. Each super-node is responsible for maintaining the related information of each member of the cluster. Finally, normal nodes are responsible for providing and sharing resources. The network resulting topology is shown in Figure 4.

There are other kinds of semi-structured P2P networks where the network is divided into different subnets, resulting in a topology based on the peers' interests. In [93] a system is presented where nodes are clustered according to their interests [33] to form a P2P overlay network of multilayer interest domains. In this system, there are three types of nodes: active nodes, super-nodes, and normal nodes. Each active node acts as a router providing information that facilitates query routing information at the cluster level. A super-node is a representative node of a cluster and is in charge of maintaining the related information of each member node in the cluster. Finally, a normal node is mainly responsible for providing various types of shared resources. The resulting network topology is shown in Figure 4. Another similar approach is presented in [133], where the authors propose a system in which clusters are constructed on multiple logical layers. In this system, peers can switch overlay networks to search content based on popularity. One of the overlay networks is a network based on clusters constructed according to the content of each peer [134,104].

Social media has changed our way of communication and sharing data on the Internet, which is now mostly based on collaboration among members to provide and exchange information. The efficiency of this new form of interaction motivates researchers to design architectures based on the social behavior of the users. In [30] an algorithm called *ROUTIL*, that combines social computing and P2P systems, is presented. The goal in ROUTIL is to link users with similar interests in order to provide a secure and effective service. A method for modeling users' interest in P2P-document-sharing systems based on k-medoids clustering is presented in [108]. In the proposed approach an overlay network is created based on the k-medoids clustering algorithm, which is combined with the users' historical queries to improve the initial user interest model.

3.3. Routing algorithms in unstructured P2P networks

In an unstructured P2P network (Figure 5), there is no specific criterion which strictly defines where data is stored and which nodes are neighbors of each other. The *Breadth First Search* (BFS) or flooding is the typical algorithm used to search in pure P2P networks. In these algorithms, queries are propagated from a node to all of its neighbors, then to the neighbors of those nodes and so on, until the TTL parameter becomes zero. This routing method is implemented in some systems such as Napster and Gnutella [100,111]. Flooding tries to find the maximum number of results. However, flooding does not scale well [111], and it generates a large number of messages in comparison with other approaches.

Many alternative schemes have been proposed to address the original flooding problems in unstructured P2P networks. In *iterative deepening* [142], also called expanding ring, the query-issuing node periodically carries out a sequence of BFS searches with increasing depth limits $D_1 < D_2 < ... < D_n$. The query is considered to be resolved when the query result is satisfied or when the maximum depth limit *n* has been reached. In the latter case, the query is assumed to remain unsolved, and it can be determined that the query-issuing node will never find the answer to that query. All nodes use the

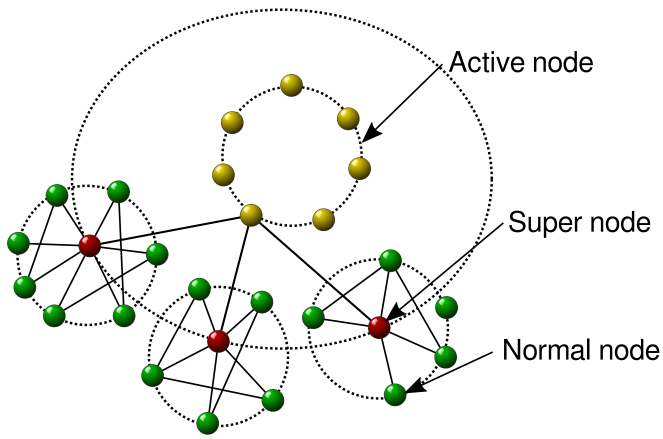

Fig. 4. Network topology structure in a P2P overlay network of multilayer interest domains (adapted from [93]).

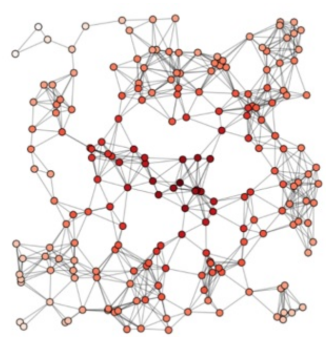

Fig. 5. Graphical representation of an unstructured topology in a P2P network.

same sequence of depth limits called *policy P* and the same period of time between two consecutive BFS searches. This algorithm is appropriate for applications where the initial number of query hits is important, but this approach does not reduce the number of duplicate messages and the associated query processing time is high.

In a *Depth-First Search* (DFS) algorithm, rather than sending a query to all the neighbors, each peer selects a single candidate neighbor to send the query. In this scheme, the maximum TTL of a query is used to specify the search depth. If the query-originating node does not receive a reply within a certain period of time, the node selects another neighbor to send the query. The process is repeated until the query is answered or all the neighbors have been selected. The criteria used to select a neighbor can highly influence the performance of the search process. FreeNet [41,4] is an example of a P2P system using the DFS scheme.

In the standard *random-walk* algorithm [55], the query-issuing node forwards the query to one neighbor selected randomly. On its turn, this neighbor proceeds in a similary way, choosing randomly one of its neighbors and forwarding the query message to that neighbor. The procedure is repeated until the required data is found. This algorithm uses only one walker, reducing the message overhead but causing longer search delays. In the *k-walker random walk* [88], k copies of the query message are sent by the query-issuing node to k randomly selected neighbors. Each query message takes its own random walk. In order to decide if a termination condition has been reached, each walker periodically communicates with the query-issuing node. This algorithm attempts to reduce the routing delay. A similar approach is the *two-level random-walk* algorithm [67]. In this algorithm, the query-issuing node uses k_1 random search threads with a TTL with a value of l_1. When this TTL parameter expires, each search thread explodes to k_2 search threads with the TTL parameter established in l_2. This approach aims to reduce duplicate messages, but it has a longer search delay than the k-walker random walk. Random-walk approaches are popular in P2P applications. For example, in [80] a study of the random-walk domination problem is presented with the formulation of an effective greedy algorithm that guarantees an optimal performance.

Another similar approach is the *modified random BFS* algorithm [71] where the query-issuing node forwards the query to a randomly selected subset of its neighbors. On receiving a query message, each neighbor forwards the query to a randomly selected subset of its neighbors (excluding the query-issuing node). This algorithm continues until some stop condition is satisfied. As pointed out in [71], this approach results in more nodes being visited and has a higher query success rate than the k-walker random walk.

Directed BFS [142] is a routing algorithm that selects those neighbors from the query-issuing node which are expected to quickly return many high-quality results. The selected neighbors subsequently forward the query message in a BFS way to all their neighbors. Each peer stores simple statistics about its neighbors (e.g. the highest number of query results returned previously, network latency for the neighbor, or the least busy neighbors) and uses this information for a more informed neighbor selection strategy.

Intelligent search [71] is similar to directed BFS. However, a more intelligent approach to neighbor selection is achieved by considering the past performance of the query-issuing node neighbors and limiting query propagation only to a selected subset of these neighbors. These neighbors are selected through a query-oriented approach that considers whether the neighbors have successfully answered similar queries (based on query cosine similarity [64]) in the past. Each node keeps a profile of its neighbors with information on those queries that the neighbors answered more recently in the past. Similarly to other query propagation approaches, a TTL value is used to stop the query propagation process.

In the *local-index-based search* algorithm [142], every node replicates the indices maintained by other nodes for their local data with a k-hop distance from it. In this way, a node can use data from its local indices to answer queries associated with data stored in other nodes. A broadcast policy P defines when query propagation must stop. As a consequence, only those nodes at depths smaller than those listed in P check their local indices and return the query result if the requested data is found. Local indices are updated to reflect changes when a node joins, leaves, or modifies its own data. At the time a node Y joins the network it sends a join message with a TTL of r hops. Hence, all nodes within an r-hop distance from Y receive this message. The message contains metadata describing Y's data collection. If a node X receives a join message from Y, it replies with another join message with metadata describing its own data collection to keep Y's index up to date. Each time a node Z leaves the network or dies, other nodes that index Z update their indices after a timeout, removing information on Z's data collections from their indices. Modifications on Z's data collections are reflected on other nodes indices by sending a short update message with a TTL of r to all Z's neighbors. Query propagation in the local-index-based search approach is similar to the iterative deepening approach in that both algorithms rely on a list of depths to limit the number of hops allowed. However, while in iterative deepening nodes maintain indices containing local information only, in local-index-based search, nodes maintain indices containing not only local information but also information about data collections from other nodes.

The *routing-index-based search* algorithm [43] is similar to directed BFS and intelligent search. The three approaches guide the entire search process using neighbor information but differ in the type of information stored and the way this information is used. In directed BFS only the query-issuing node uses this information to select appropriate peers while the rest of the nodes use BFS as a strategy to route queries. Intelligent search uses information about the past queries that have been answered by neighbors. However, different from the rest, routing-index-based search stores information about the number of documents and the topics of the documents stored in the neighbor nodes. This facilitates the process of selecting the best candidate neighbors to forward queries. In routing-index-based search, good neighbors typically provide a means to quickly find many documents. Since indices are required to be small in a distributed-index mechanism routing indices do not maintain the location of each document. Instead they maintain information to guide the process of finding a document.

To illustrate the routing indices approach, we revisit the example presented in [43]. Consider Figure 6 which shows four nodes A, B, C, and D, connected by solid lines. The document with content x is located at node C, but the RI of node A points to neighbor B instead of pointing directly to C (dotted arrow). By using "routes" rather than destinations, the indices are proportional to the number of neighbors, rather than to the number of documents. The size of the RIs is reduced by using approximate indices, i.e., by allowing RIs to give a hint (rather than a definite answer) about the location of a document. For example, in the same figure, an entry in the RI of node A may cover documents with contents x, y or z. A request for documents with content x will yield a correct hint, but one for content y or z will not. This is a content-oriented method such that the node knowledge about topics belonging to other peers is updated when a node establishes a new connection (not by past experience).

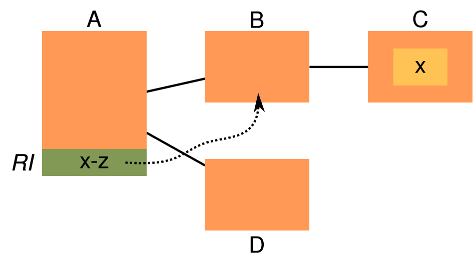

Fig. 6. Routing Indices schema (adapted from [43]).

Some P2P information retrieval methods are adaptations of a classification problem to query routing. In a classification problem, the classifier tries to classify an object using some features. The *Semantic Overlay Model* [68] aims at locating appropriate peers to answer a specific query. Instead of broadcasting queries, this approach routes queries to semantically similar peers. They produce semantic vectors in order to classify peers into categories that represent the peers' semantic similarity. This is a content-oriented method given that it uses meta-information to classify peers by interests. A query can be routed to related peers, increasing the recall rate while reducing at the same time hops and messages. Experiments have shown that establishing a semantic overlay model based on latent semantic indexing and support vector machine methods is feasible and performs well and that the query routing algorithm in the semantic overlay is efficient [68] .

In a pure P2P unstructured and decentralized network all the peers usually have the same responsibilities. However, some query-routing methods in unstructured P2P network make use the notion of "super-peer" as is the case in the *Backpressure* algorithm [118]. In this algorithm, super-peers serve their subordinates by resolving queries or forwarding them to other super-peers. Super-peers can resolve queries by checking the files/resources they have, as well as those of their subordinate community. Methods that impose some structure on special peers are considered "routing algorithms for semi-structured P2P networks", but in this case super-peers are self-organized without a central or initial control. The algorithm *Backpressure* is query-oriented, and it uses past information to decide how to route queries disregarding the content of each peer.

The *Route Learning* algorithm [40] uses keywords extracted from queries to determine how to route the queries. This differs from other approaches, where meta-data is used to classify queries and to decide how to route them. In the Route Learning scheme, a peer tries to estimate the most likely neighbors to reply to queries. Peers calculate this estimation based on the knowledge that is gradually built from query and query hit messages sent to and received from the neighbors. Route Learning reduces the query overhead in flooding-based networks using keywords extracted from queries, being therefore a query-oriented method.

Routing future queries using past experiences is the best way to route queries to specific nodes with the objective of improving performance, but it is important to store this accumulated knowledge in an efficient way. As a consequence, each peer may need to consider some storage space for maintaining metadata. This storage also implies the cost of keeping track of these data updates. The *Learning Peer Selection* approach [26] implements a query-oriented method on unstructured networks with the ability to discover users' preferences by analyzing their download history. The proposed model is implemented in three layers. The first of these layers is especially dedicated to store and to update past information. The other two layers are responsible for managing users' profiles and selecting the relevant peers to send queries.

Cooperation among peers in a P2P network is strongly linked with the concept of knowledge sharing. Usually, there is a trade-off between improving the network global knowledge and the cost of sending update messages through the network. The *Self Learning Query Routing* algorithm [38] attempts to improve global knowledge by learning the nodes' interests based on their past search result history. The number of shared files determines a rank of friendship between two nodes. Queries are initially routed to friend nodes only. In case of failure, a broadcast search is executed. Past search results allow nodes to incrementally learn about other nodes in the network that share the same interests.

A P2P algorithm that relies on the notion of semantic communities is *INGA* [83]. The INGA algorithm assumes that each peer plays a different role in a social network , such as content provide, recommender, etc. The roles associated with peers allow INGA to determine the best matching candidates to which a query should be forwarded. Facts are stored and managed locally on each peer, constituting the *topical knowledge* of the peer. Each peer maintains a personal semantic shortcut index. An evaluation of different P2P search strategies based on the *6S* system [139] is carried out in [22] with the purpose of showing the emergence of semantic communities. The query-routing evaluated strategies include a *random*, a *greedy* and a *reinforcement learning* algorithm. To route queries appropriately, in the greedy and the reinforcement learning algorithms, each peer learns and stores profiles of other peers. A neighbor profile is defined by the information that a peer maintains in order to describe the contents stored by a given neighbor. By adapting the profile information, peers try to increase the probability of choosing the appropriate neighbors for their queries. Simulations demonstrate that peers can learn from their interactions to form semantic communities even when the overlay network is unstructured. Another content-oriented search algorithm is *State-based search* (SBS) [138]. In this algorithm, each node maintains a list with state information associated with the other nodes in the network and uses this information to route queries. Searches are performed using a local fuzzy logic-based routing algorithm. Results reported by the authors indicate that SBS reduces the response time and obtains a better load balance when compared to baseline algorithms.

An approach aimed at achieving low bandwidth is *Scalable Query Routing* (SQR) [76]. In this algorithm, a routing table is maintained at each node that suggests the location of objects in the network based on the past experience. A data structure called *Exponentially Decaying Bloom Filter* (EDBF) encodes probabilistic routing tables in a highly compressed manner and allows efficient query propagation. Other content-oriented methods seek to control the system congestion by tracking alternative routes to balance the query load between peers. For instance, the method presented in [120] relies on a *Collaborative Q-Learning* algorithm that learns several parameters associated with the network state and performance. In [31] two algorithms are presented that are a combination of other existing techniques. One algorithm is a combination of *Flooding* and *Random Walk* while the other combines *Flooding* with *Random Walk* with Neighbors Table. The authors present different results obtained from simulations over an unstructured P2P network that showed that hybrid algorithms provide the most balanced performance regarding the average number of hops, average search time and the number of failures when compared to the basic resource discovery algorithms.

Other relevant search algorithms in unstructured P2P networks that are not described in this article but are classified in the following section are *q-pilot* [126], *SemAnt* [95], *Remindin'* [128], *P2PSLN* [152] and *NeuroGrid* [69].

4. Comparative Analysis

Next we will present a comparative analysis of the major features involved in the query routing process. We will also discuss the advantages and disadvantages of the different existing approaches.

4.1. Features comparison

In this subsection, a comparative analysis of the algorithms previously described is presented. Table 1 shows a comparison between routing algorithms in structured P2P networks. In this kind of systems, the use of DHT allows ensuring a logarithmic execution time. The algorithm used by the *SkipNet* and *Small-Word* scheme shows a central difference with respect to the other algorithms presented in table 1. In Chord and Pastry, the goal is to implement a DHT diffusing content randomly throughout an overlay in order to obtain a uniform and load-balanced behavior, whereas in *SkipNet* the goal is to enable systems to preserve useful content and path locality using the Small-World topology to take advantage of shortcuts to remote nodes.

Table 1. Comparison of salient features that characterize structured P2P networks.

		Features			
		DHT	Overlay		Structure
			Flat	Hierarchical	
Algorithm	Chord	•	•		Ring
	Pastry	•	•		Tree
	Tapestry	•	•		Tree
	CAN	•	•		Toroidal
	KaZaA	•		•	2-layers
	SkipNet and Small World			•	2-layers
	pSearch	•		•	Toroidal
	AFT		•		Toroidal
	HBFR			•	Geographical
	OntoSum	•		•	Tree

In unstructured P2P networks, search turns out to be a difficult, non-scalable process [75]. As a consequence, these algorithms take advantage of different semantic aspects in order to optimize their associated search processes. Table 2 outlines the semantic aspects that are present in each of the unstructured systems described above. The first five are flooding-like algorithms, and consequently they do not consider any semantic aspect. In the rest of the algorithms, the goal is to strategically select candidate nodes in order to reduce query propagation. To do that, some algorithms (e.g. Directed BFS) use heuristic information, while others select the candidate nodes by past experience (query-oriented) or by analyzing the profile of a node (content-oriented). Finally, there is a subset of algorithms that use a classifier to decide which are the best candidate nodes.

Table 2. Semantic aspects in unstructured P2P networks.

		Semantic Aspects			
		Heuristic Information	Content Oriented	Query Oriented	Classification
Algorithm	Flooding				
	Iterative Deeping				
	Random Walk				
	K-walker Random Walk				
	Two-level K-walker Random Walk				
	Modified Random BFS				
	Directed BFS	•		•	
	Intelligent Search			•	
	Local Indices Based Search	•		•	
	Routing Indices Based Search		•		
	Semantic Overlay Model		•		•
	Route Learning			•	•
	Learning Peer Selection			•	
	Self Learning Query Routing			•	
	6S - Random				
	6S - Greedy		•		
	6S - Reinforcement Learning		•		
	q-pilot				•
	SemAnt	•			•
	REMINDIN'			•	
	P2PSLN		•		
	NeuroGrid		•		
	INGA		•		
	SQR	•			
	SBS		•		
	Collaborative Q-Learning	•	•		

There are some algorithms (such as BFS, DFS, and random approaches) that do not exploit semantic aspects and consequently they are forced to implement a less informed method for propagating queries. These features can be observed in table 3. From this table, we can appreciate that even those algorithms that account for semantic aspects have a basic mechanism to propagate queries. These mechanisms are executed over the subset of candidate nodes or when no candidate node exists and queries must be propagated in an alternative way. Another feature that is present in this kind of algorithms is the TTL parameter, which is decremented by one every time the message goes from a node to another. By performing this process, when the value of the TTL parameter becomes zero the search for candidates can be assumed to be over, so that the original message can be ultimately discarded.

Table 3. Basic routing algorithms in unstructured P2P networks.

		Features			
		BFS	DFS	Random	TTL
Algorithm	Flooding	●			●
	Iterative Deeping	●			●
	Random Walk	●		●	●
	K-walker Random Walk	●		●	
	Two-level K-walker Random Walk	●		●	●
	Modified Random BFS	●		●	n/a
	Directed BFS	●			n/a
	Intelligent Search	●			●
	Local Indices Based Search	●			●
	Routing Indices Based Search		●		n/a
	Semantic Overlay Model	n/a	n/a	n/a	●
	Route Learning	n/a	n/a	n/a	●
	Learning Peer Selection	n/a	n/a	n/a	●
	Self Learning Query Routing	●			n/a
	6S - Random	●		●	●
	6S - Greedy	●			●
	6S - Reinforcement Learning	●			●
	q-pilot	n/a	n/a	n/a	●
	SemAnt	n/a	n/a	n/a	●
	REMINDIN’	n/a	n/a	n/a	n/a
	P2PSLN	n/a	n/a	n/a	●
	NeuroGrid	●			●
	INGA	n/a	n/a	n/a	n/a
	SQR	n/a	n/a	n/a	n/a
	SBS	n/a	n/a		●
	Collaborative Q-Learning	●			●

Figure 7 presents a timeline that shows the evolution of the main routing algorithms in P2P networks over the years. From this timeline we can see that between 2002 and 2005 there was a considerable growth of algorithms for unstructured networks, whereas in recent years research efforts have been particularly focused on hybrid and structured networks. Furthermore, it can be seen that among algorithms for searching in unstructured networks, intelligent algorithms are still a minority, being most of them based on basic flooding mechanisms.

As introduced in Section 2, query routing algorithms can be classified according to the topology of the underlying network. Algorithms for structured networks use some data structure in order to select destination peers. Most of these algorithms use data structures such as trees or rings, but there are other less usual structures such as the geographical position of the peers or toroidal. Algorithms for query routing in unstructured P2P networks can be classified according to the degree of intelligence that they use to select destination peers. Most of these algorithms are based on basic routing techniques using a random parameter or simply based on graph paths. Intelligent algorithms use different strategies for routing queries, which range from the adoption of particular classification techniques to the use of different kinds of heuristics. Figure 8 shows the main features present in structured/unstructured routing algorithms.

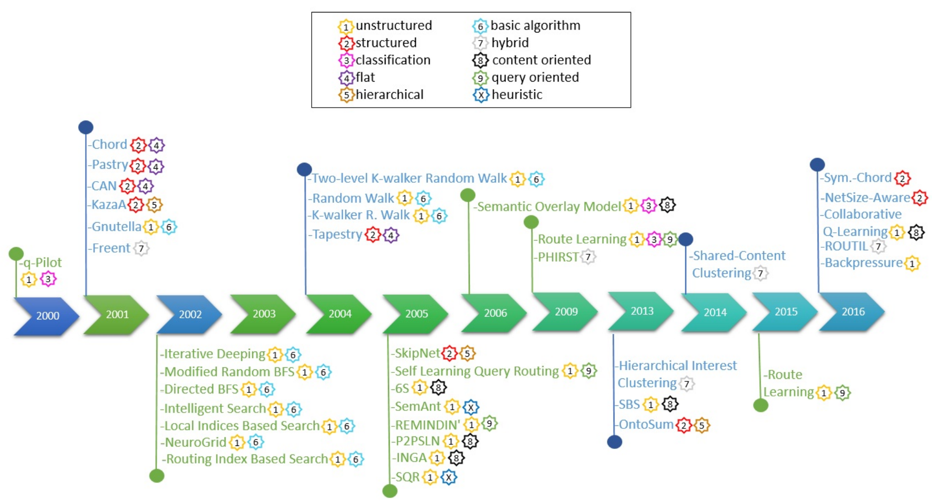

Fig. 7. Evolution of the main routing algorithms.

4.2. Discussion

P2P systems are distributed systems consisting of interconnected nodes that offer support to different applications such file sharing, distributed data storage, and distributed social networks, among others. Developing reliable, robust, effective and efficient P2P systems gives rise to research challenges such as the design of reputation systems in P2P

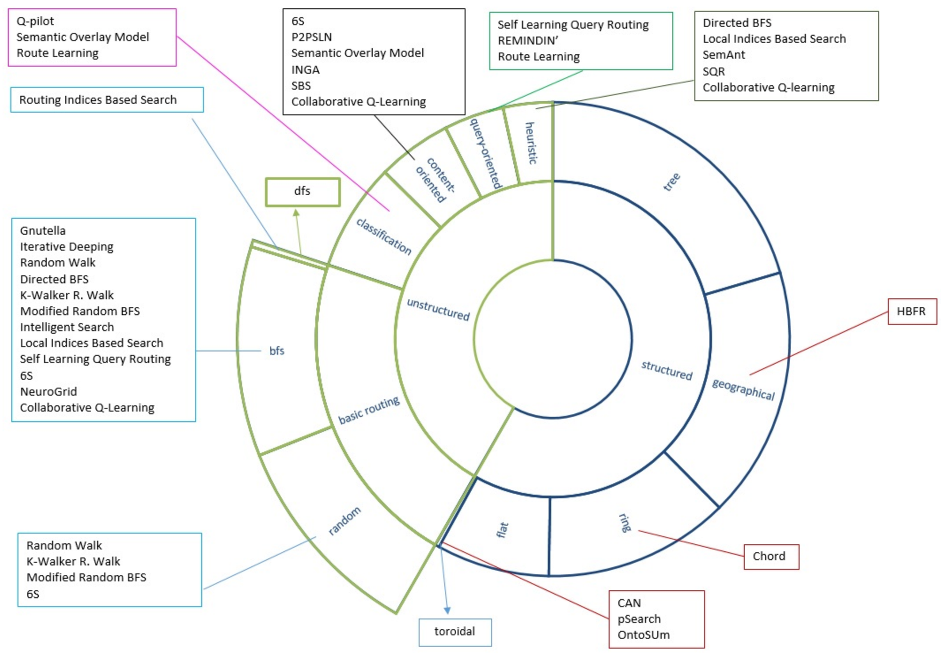

Fig. 8. Salient features of structured/unstructured routing algorithms .

environments, the exploitation of the semantic organization of information, the use of cryptographic mechanisms for data protection, etc. Several design features commonly considered when developing P2P systems include:

- **Replication.** P2P systems rely on content replication to ensure content availability. Replication is a major challenge for structured systems such as *Chord*, where identifiers are linked to their location. In these cases alias are used to allow replication [124]. *CAN* utilizes a replica function to produce random keys to store copies at different locations [136].
- **Security.** The dynamic and autonomous nature of P2P systems poses several challenges at the moment of ensuring availability, privacy, confidentiality, integrity, and authenticity [25]. Security in P2P networks is a highly explored field. Several cryptography algorithms and protocols have been developed especially for P2P systems, such as *Self-Certifying Data* and *Information Dispersal* [37,109]. Security also involves detecting and managing malicious nodes that can corrupt messages that are propagated among other nodes. The "Sybil Attack" [47] is a security threat related to authenticity where a node in a network claims multiple identities. The *Symmetric Chord* routing algorithm addresses the authenticity problem by implementing an authenticity validation process. Other approaches to address security issues in P2P networks rely on the use of special forms of access control lists [45].
- **Anonymity.** Author anonymity or peer anonymity is some times required in P2P applications. An approach named "Disassociation of Content Source and Requestor"

adopted by *FreeNet* provides anonymity to users by preventing other nodes from discovering the true origin of a file in the network. Another anonymity mechanism is "Censorship Resistant Lookup", which is used by *Achord* [60], a variant of the *Chord* lookup service.

– **Incentive mechanisms.** The performance of a decentralized P2P system relies on the voluntary participation of its users. To achieve a good performance it is necessary to implement methods that provide incentives to stimulate cooperation among users [58]. A simple incentive mechanism is based on ranking highly the results of a particular node if it has contributed significantly in previous searches. This simple method typically works for both the Web and P2P networks since appearing high up in a ranking typically represents an incentive for companies and people [42].
– **Semantic grouping of information.** The semantic organization of content through the emergence of semantic communities is deeply analyzed in [22]. Some systems are based on the notion of "peer communities", where relationships among peers are based on nodes' interests [73]. The emergence of these communities tends to make the search process more effective as it is possible to target search queries to those communities more closely related to the topic of the queries. This feature is exploited by some routing algorithms as discussed in [22,83].

There are several systems that use P2P technologies, such as those presented in section 2.5. These systems adopted different architectures and routing algorithms. We consider that no architecture is better than another, but can be use in deferents contexts. While a structured architecture can guarantee a determined execution time, a decentralized one can adapt more easily to topology changes. Decentralized approaches need to store some data to decide how to route a query but most of the algorithms for structured topologies need to keep their DHT update. Finally, as P2P technologies are still evolving, there are some open research problems such as a) developing routing algorithms for maximizing performance; b) defining more efficient security, anonymity, and censorship resistance schemes; c) exploiting semantic grouping of information in P2P networks; d) developing more effective incentive mechanisms.

5. Conclusions

The early Internet was designed on principles of cooperation and good engineering. In many ways, it shared principles and concepts with pure P2P networks. In this decentralized scenario, algorithms that performed searches were essential. When Internet became more rigid, structured and semi-structured search algorithms emerged, where collaboration among peers was no longer an important issue. Nevertheless, in the last few years pure P2P networks have come back for playing a major role in the deployment of new systems and technologies. Research in decentralized search has given rise to novel algorithms that incorporate semantic aspects derived from the profile of each participant. These semantic aspects can be conveniently exploited to improve routing algorithms, with the goal of minimizing network traffic and optimizing query response time.

In this article, we have reviewed the most important query routing algorithms in P2P networks, contrasting their advantages and disadvantages. To facilitate the analysis of these algorithms, we have introduced different schemes and classifications. In particular, we have discussed diverse search strategies in structured, semi-structured, and unstructured P2P networks. Finally, we have identified common features in these networks, carrying out a comparative analysis for contrasting these features.

As discussed in this article, semantic issues in intelligent query routing provide a significant added value for improving search in distributed environments. This survey aims at offering an in-depth analysis of the state of the art in this exciting research area, oriented towards a wide and heterogeneous audience of researchers and practitioners working on P2P networks. New recent advances in Artificial Intelligence techniques (e.g. [103]) show that future developments in P2P networks will allow to go beyond the traditional semantic analysis by adding qualitative reasoning capabilities to the nodes. Even though some motivating preliminary results have been obtained, most of the research work in this direction is still to be done.

Acknowledgements

We want to thank the reviewers who provided helpful suggestions and insights for improving the original version of the article. This research was supported by the projects PICT-ANPCyT 2014-0624, PIP-CONICET 112-2012010-0487, and PGI-UNS 24/N039.

References

1. Kazaa. `http://www.kazaa.com`. Retrieved in September 2011 (2011)
2. Buddycloud. `http://buddycloud.com/`. Retrieved in June 2017 (2017)
3. Diaspora. `https://diasporafoundation.org/`. Retrieved in June 2017 (2017)
4. Freenet. `http://freenetproject.org`. Retrieved in June 2017 (2017)
5. Friendica. `http://friendi.ca/`. Retrieved in June 2017 (2017)
6. Gnusocial. `https://gnu.io/social/`. Retrieved in June 2017 (2017)
7. Kune. `http://kune.ourproject.org`. Retrieved in June 2017 (2017)
8. Mastodon. `https://mastodon.social`. Retrieved in June 2017 (2017)
9. Minds. `https://www.minds.com/`. Retrieved in June 2017 (2017)
10. Twister. `http://twister.net.co/`. Retrieved in June 2017 (2017)
11. Apache cassandra. `http://cassandra.apache.org`. Retrieved in September 2018 (2018)
12. Bigtable. `http://cloud.google.com/bigtable/`. Retrieved in September 2018 (2018)
13. Dynamo. `http://aws.amazon.com/es/dynamodb/`. Retrieved in September 2018 (2018)
14. emule. `www.emule-project.net`. Retrieved in September 2018 (2018)
15. Facebook. `http://facebook.com/`. Retrieved in September 2018 (2018)
16. Myspace. `http://myspace.com/`. Retrieved in September 2018 (2018)
17. Popcorn time. `https://popcorntime.sh/es`. Retrieved in September 2018 (2018)
18. Twitter. `http://twitter.com/`. Retrieved in September 2018 (2018)
19. Windows azure storage. `http://azure.microsoft.com`. Retrieved in September 2018 (2018)
20. Aberer, K., Hauswirth, M.: An overview of peer-to-peer information systems. In: Workshop on Distributed Data and Structures. vol. 14, pp. 171–188 (2002)
21. Adamic, L.A., Lukose, R.M., Puniyani, A.R., Huberman, B.A.: Search in power-law networks. Physical Review E 64, 046135 (Sep 2001)

22. Akavipat, R., Wu, L.S., Menczer, F., Maguitman, A.G.: Emerging semantic communities in peer web search. In: Proceedings of the international workshop on Information retrieval in peer-to-peer networks. pp. 1–8. P2PIR '06, ACM, New York, NY, USA (2006)
23. Amad, M., Aïssani, D., Meddahi, A., Benkerrou, M., Amghar, F.: De bruijn graph based solution for lookup acceleration and optimization in p2p networks. Wireless Personal Communications 85(3), 1471–1486 (Dec 2015), https://doi.org/10.1007/s11277-015-2851-y
24. Androutsellis-Theotokis, S., Spinellis, D.: A survey of peer-to-peer content distribution technologies. ACM Computing Surveys (CSUR) 36(4), 335–371 (2004)
25. Androutsellis-Theotokis, S., Spinellis, D.: A survey of peer-to-peer content distribution technologies. ACM Comput. Surv. 36(4), 335–371 (Dec 2004), http://doi.acm.org/10.1145/1041680.1041681
26. Arour, K., Yeferny, T.: Learning model for efficient query routing in P2P information retrieval systems. Peer-to-Peer Networking and Applications 8(5), 741–757 (2015)
27. Azimi, R., Sajedi, H., Ghayekhloo, M.: A distributed data clustering algorithm in p2p networks. Applied Soft Computing 51, 147–167 (2017)
28. Babaei, H., Fathy, M., Romoozi, M.: Modeling and optimizing random walk content discovery protocol over mobile ad-hoc networks. Performance Evaluation 74, 18–29 (2014)
29. Baccelli, F., Mathieu, F., Norros, I., Varloot, R.: Can P2P networks be super-scalable? In: INFOCOM 2013. Annual Joint Conference of the IEEE Computer and Communications Societies. pp. 1753–1761. IEEE (2013)
30. Badis, L., Amad, M., Aîssani, D., Bedjguelal, K., Benkerrou, A.: Routil: P2p routing protocol based on interest links. In: Advanced Aspects of Software Engineering (ICAASE), 2016 International Conference on. pp. 1–5. IEEE (2016)
31. Bashmal, L., Almulifi, A., Kurdi, H.: Hybrid resource discovery algorithms for unstructured peer-to-peer networks. Procedia Computer Science 109, 289–296 (2017)
32. Bawa, M., Manku, G.S., Raghavan, P.: Sets: search enhanced by topic segmentation. In: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval. pp. 306–313. ACM (2003)
33. Ben-Gal, I., Shavitt, Y., Weinsberg, E., Weinsberg, U.: Peer-to-peer information retrieval using shared-content clustering. Knowledge and information systems 39(2), 383–408 (2014)
34. Brienza, S., Cebeci, S.E., Masoumzadeh, S.S., Hlavacs, H., Özkasap, Ö., Anastasi, G.: A survey on energy efficiency in p2p systems: File distribution, content streaming, and epidemics. ACM Computing Surveys (CSUR) 48(3), 36 (2016)
35. Broder, A., Mitzenmacher, M.: Network applications of bloom filters: A survey. Internet mathematics 1(4), 485–509 (2004)
36. Castano, S., Montanelli, S.: Semantic self-formation of communities of peers. In: Workshop on Ontologies in Peer-to-Peer Communities. European Semantic Web Conference (2005)
37. Castro, M., Druschel, P., Ganesh, A., Rowstron, A., Wallach, D.S.: Secure routing for structured peer-to-peer overlay networks. ACM SIGOPS Operating Systems Review 36(SI), 299–314 (2002)
38. Chen, H., Gong, Z., Huang, Z.: Self-learning routing in unstructured P2P network. International Journal of Information Technology 11(12), 59–67 (2005)
39. Choi, J., Han, J., Cho, E., Kwon, T.T., Choi, Y.: A survey on content-oriented networking for efficient content delivery. Communications Magazine 49(3), 121–127 (2011)
40. Ciraci, S., Körpeoglu, I., Ulusoy, O.: Reducing query overhead through route learning in unstructured peer-to-peer network. Journal of Network and Computer Applications 32(3), 550–567 (May 2009)
41. Clarke, I., Sandberg, O., Wiley, B., Hong, T.W.: Freenet: A distributed anonymous information storage and retrieval system. In: Designing Privacy Enhancing Technologies. pp. 46–66. Springer (2001)

42. Craswell, N., Hawking, D.: Web information retrieval, chap. 5, pp. 85–101. John Wiley & Sons, Ltd (2009)
43. Crespo, A., Garcia-Molina, H.: Routing indices for peer-to-peer systems. In: Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS'02). pp. 23–32. IEEE Computer Society (2002)
44. Crespo, A., Garcia-Molina, H.: Semantic overlay networks for P2P systems. In: Agents and Peer-to-Peer Computing, pp. 1–13. Springer (2005)
45. Daswani, N., Garcia-Molina, H., Yang, B.: Open problems in data-sharing peer-to-peer systems. In: Database Theory–ICDT 2003, pp. 1–15. Springer (2003)
46. de Bruijn, N.: A combinatorial problem. Proceedings of the Koninklijke Nederlandse Akademie van Wetenschappen. Series A 49(7), 758–764 (1946)
47. Douceur, J.R.: The sybil attack. In: International workshop on peer-to-peer systems. pp. 251–260. Springer (2002)
48. Du, A., Callan, J.: Probing a collection to discover its language model. Tech. rep., University of Massachusetts (1998)
49. Dunn, R.J., Zahorjan, J., Gribble, S.D., Levy, H.M.: Presence-based availability and P2P systems. In: Peer-to-Peer Computing, 2005. P2P 2005. Fifth IEEE International Conference on. pp. 209–216. IEEE (2005)
50. Einhorn, M.A., Rosenblatt, B.: Peer-to-peer networking and digital rights management: How market tools can solve copyright problems. J. Copyright Soc'y USA 52, 239 (2004)
51. Fanti, G., Viswanath, P.: Anonymity properties of the bitcoin p2p network. arXiv (2017), https://arxiv.org/abs/1703.08761
52. Felber, P., Kropf, P., Schiller, E., Serbu, S.: Survey on load balancing in peer-to-peer distributed hash tables. IEEE Communications Surveys and Tutorials 16(1), 473–492 (2014)
53. Fraigniaud, P., Gauron, P.: D2b: A de bruijn based content-addressable network. Theoretical Computer Science 355(1), 65 – 79 (2006), http://www.sciencedirect.com/science/article/pii/S0304397505009163, complex Networks
54. Galluccio, L., Morabito, G., Palazzo, S., Pellegrini, M., Renda, M.E., Santi, P.: Georoy: A location-aware enhancement to viceroy peer-to-peer algorithm. Computer Networks 51(8), 1998–2014 (2007), https://doi.org/10.1016/j.comnet.2006.09.017
55. Gkantsidis, C., Mihail, M., Saberi, A.: Random walks in peer-to-peer networks. In: INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies. vol. 1. IEEE (2004)
56. Gravano, L., Chang, K., Garcia-Molina, H., Paepcke, A.: Starts: Stanford protocol proposal for internet retrieval and search. Tech. rep., Stanford University, Stanford, CA, USA (1997)
57. Grumbach, S., Riemann, R.: Secure and trustable distributed aggregation based on kademlia. In: di Vimercati, S.D.C., Martinelli, F. (eds.) ICT Systems Security and Privacy Protection - 32nd IFIP TC 11 International Conference, SEC 2017, Rome, Italy, May 29-31, 2017, Proceedings. IFIP Advances in Information and Communication Technology, vol. 502, pp. 171–185. Springer (2017), https://doi.org/10.1007/978-3-319-58469-0_12
58. Haddi, F.L., Benchaïba, M.: A survey of incentive mechanisms in static and mobile P2P systems. Journal of Network and Computer Applications 58, 108–118 (2015)
59. Han, J., Haihong, E., Le, G., Du, J.: Survey on nosql database. In: Pervasive computing and applications (ICPCA), 2011 6th international conference on. pp. 363–366. IEEE (2011)
60. Hazel, S., Wiley, O.: Achord: A variant of the chord lookup service for use in censorship resistant peer-to-peer publishing systems (04 2002)
61. Heck, H., Kieselmann, O., Wacker, A.: Evaluating connection resilience for the overlay network kademlia. In: Lee, K., Liu, L. (eds.) 37th IEEE International Conference on Distributed Computing Systems, ICDCS 2017, Atlanta, GA, USA, June 5-8, 2017. pp. 2581–2584. IEEE Computer Society (2017), https://doi.org/10.1109/ICDCS.2017.101
62. Herschel, S.: Indexing dynamic networks. In: Cremers, A.B., Manthey, R., Martini, P., Steinhage, V. (eds.) Lecture Notes in Informatics. vol. 65, pp. 429–433 (2005)

63. Hsu, C.Y., Wang, K., Shih, H.C.: Decentralized structured peer-to-peer network and load balancing methods thereof (May 2013), uS Patent 8,443,086
64. Huang, A.: Similarity measures for text document clustering. In: Proceedings of the Sixth New Zealand Computer Science Research Student Conference (NZCSRSC2008). pp. 49–56 (2008)
65. Huang, X., Chen, L., Huang, L., Li, M.: Routing algorithm using skipnet and small-world for peer-to-peer system. In: Proceedings of the 4th International Conference on Grid and Cooperative Computing. pp. 984–989. GCC'05, Springer-Verlag, Berlin, Heidelberg (2005)
66. Ismail, H., Germanus, D., Suri, N.: P2p routing table poisoning: A quorum-based sanitizing approach. Computers & Security 65, 283–299 (2017)
67. Jawhar, I., Wu, J.: A two-level random walk search protocol for peer-to-peer networks. In: 8th World Multi-Conference on Systemics, Cybernetics and Informatics. pp. 1–5 (2004)
68. Jin, H., Ning, X., Chen, H., Yin, Z.: Efficient query routing for information retrieval in semantic overlays. In: 21st Annual ACM Symposium on Applied Computing (SAC'06). pp. 23–27. ACM Press (2006)
69. Joseph, S.: NeuroGrid: Semantically Routing Queries in Peer-to-Peer Networks. In: Gregori, E., Cherkasova, L., Cugola, G., Panzieri, F., Picco, G. (eds.) Web Engineering and Peer-to-Peer Computing, Lecture Notes in Computer Science, vol. 2376, pp. 202–214. Springer Berlin Heidelberg (2002)
70. Kaashoek, M.F., Karger, D.R.: Koorde: A simple degree-optimal distributed hash table. In: Kaashoek, M.F., Stoica, I. (eds.) Peer-to-Peer Systems II. pp. 98–107. Springer Berlin Heidelberg, Berlin, Heidelberg (2003)
71. Kalogeraki, V., Gunopulos, D., Zeinalipour-Yazti, D.: A local search mechanism for peer-to-peer networks. In: Proceedings of the Eleventh International Conference on Information and Knowledge Management. pp. 300–307. CIKM '02, ACM, New York, NY, USA (2002)
72. Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H.: The eigentrust algorithm for reputation management in P2P networks. In: Proceedings of the 12th International Conference on World Wide Web. pp. 640–651. WWW '03, ACM, New York, NY, USA (2003)
73. Khambatti, M., Ryu, K.D., Dasgupta, P.: Structuring peer-to-peer networks using interest-based communities. In: International Workshop On Databases, Information Systems, and Peer-to-Peer Computing. pp. 48–63. Springer (2003)
74. Klampanos, I., Jose, J.: An evaluation of a cluster-based architecture for peer-to-peer information retrieval. Lecture Notes in Computer Science 4653, 380–391 (2007), `http://eprints.gla.ac.uk/39573/`
75. Kleinberg, J.: Complex networks and decentralized search algorithms. In: Proceedings of the International Congress of Mathematicians (ICM). vol. 3, pp. 1019–1044 (2006)
76. Kumar, A., Xu, J., Zegura, E.W.: Efficient and scalable query routing for unstructured peer-to-peer networks. In: INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. vol. 2, pp. 1162–1173. IEEE (2005)
77. Kurose, J.F., Ross, K.: Computer Networking: A Top-Down Approach Featuring the Internet. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edn. (2002)
78. Lele, N., Wu, L.S., Akavipat, R., Menczer, F.: Sixearch.org 2.0 peer application for collaborative web search. In: Proceedings of the 20th ACM Conference on Hypertext and Hypermedia. pp. 333–334. HT '09, ACM, New York, NY, USA (2009)
79. Li, J., Khan, S.U., Ghani, N.: Semantics-Based Resource Discovery in Large-Scale Grids, pp. 409–430. John Wiley & Sons, Inc. (2013)
80. Li, R.H., Yu, J.X., Huang, X., Cheng, H.: Random-walk domination in large graphs. In: 30th International Conference on Data Engineering (ICDE). pp. 736–747. IEEE (2014)
81. Li, X., Wu, J.: Searching Techniques in Peer-to-Peer Networks, chap. 37, pp. 617–642. Auerbach Publications, Boston, MA, USA (2006)
82. Loach, S., Bowman, D.: Heuristics-based peer to peer message routing (May 20 2008), uS Patent 7,376,749

83. Löser, A., Staab, S., Tempich, C.: Semantic methods for P2P query routing. In: Multiagent System Technologies, pp. 15–26. Springer (2005)
84. Lu, J., Callan, J.: Full-text federated search of text-based digital libraries in peer-to-peer networks. Information Retrieval 9(4), 477–498 (2006)
85. Lu, J., Callan, J.: Content-based peer-to-peer network overlay for full-text federated search. In: Large Scale Semantic Access to Content (Text, Image, Video, and Sound). pp. 490–509. RIAO '07, Le Centre De Hautes Etudes Internationales D'informatique Documentaire, Paris, France (2007), http://dl.acm.org/citation.cfm?id=1931390.1931438
86. Lua, E.K., Crowcroft, J., Pias, M., Sharma, R., Lim, S.: A survey and comparison of peer-to-peer overlay network schemes. Communications Surveys & Tutorials 7(2), 72–93 (Apr 2005)
87. Luo, B., Jin, Y., Luo, S., Sun, Z.: A symmetric lookup-based secure p2p routing algorithm. KSII Transactions on Internet & Information Systems 10(5), 2203–2217 (2016)
88. Lv, Q., Cao, P., Cohen, E., Li, K., Shenker, S.: Search and replication in unstructured peer-to-peer networks. In: Proceedings of the 16^{th} international conference on Supercomputing. pp. 84–95. ACM (2002)
89. Malatras, A.: State-of-the-art survey on P2P overlay networks in pervasive computing environments. Journal of Network and Computer Applications 55, 1–23 (2015), http://www.sciencedirect.com/science/article/pii/S1084804515000879
90. Malkhi, D., Naor, M., Ratajczak, D.: Viceroy: a scalable and dynamic emulation of the butterfly. In: Ricciardi, A. (ed.) Proceedings of the Twenty-First Annual ACM Symposium on Principles of Distributed Computing, PODC 2002, Monterey, California, USA, July 21-24, 2002. pp. 183–192. ACM (2002), http://doi.acm.org/10.1145/571825.571857
91. Maymounkov, P., Mazières, D.: Kademlia: A peer-to-peer information system based on the XOR metric. In: Druschel, P., Kaashoek, M.F., Rowstron, A.I.T. (eds.) Peer-to-Peer Systems, First International Workshop, IPTPS 2002, Cambridge, MA, USA, March 7-8, 2002, Revised Papers. Lecture Notes in Computer Science, vol. 2429, pp. 53–65. Springer (2002), https://doi.org/10.1007/3-540-45748-8_5
92. Menczer, F., Wu, L.S., Akavipat, R.: Intelligent peer networks for collaborative web search. AI Magazine 29(3), 35 (2008)
93. Meng, F., Ding, L., Peng, S., Yue, G.: A P2P network model based on hierarchical interest clustering algorithm. Journal of Software 8(5), 1262–1267 (May 2013)
94. Meng, X.: speedtrust: a super peer-guaranteed trust model in hybrid p2p networks. The Journal of Supercomputing pp. 1–28 (2018)
95. Michlmayr, E., Graf, S., Siberski, W., Nejdl, W.: Query routing with ants. In: Workshop on Ontologies in Peer-to-Peer Communities. European Semantic Web Conference (2005)
96. Morr, D.: Lionshare: A federated p2p app. In: Internet2 members meeting (2007)
97. Nah, F.F.H.: A study on tolerable waiting time: how long are web users willing to wait? Behaviour & Information Technology 23(3), 153–163 (2004)
98. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2008)
99. Naor, M., Wieder, U.: Novel architectures for p2p applications: the continuous-discrete approach. ACM Transactions on Algorithms (TALG) 3(3), 34 (2007)
100. Napster: http://free.napster.com (2011)
101. Nascimento, M.A.: Peer-to-peer: Harnessing the power of disruptive technologies. ACM SIGMOD Record 32(2), 57–58 (Jun 2003)
102. Nicolini, A.L., Lorenzetti, C.M., Maguitman, A.G., Chesñevar, C.I.: Intelligent algorithms for reducing query propagation in thematic P2P search. In: Anales del XIX Congreso Argentino de Ciencias de la Computación (CACIC). pp. 71–79. Mar del Plata, Buenos Aires, Argentina (Oct 2013)
103. Nicolini, A.L., Maguitman, A.G., Chesñevar, C.I.: Argp2p: An argumentative approach for intelligent query routing in P2P networks. In: Theory and Applications of Formal Argumentation - Third International Workshop, TAFA 2015, Buenos Aires, Argentina, July 25-26, 2015, Revised Selected Papers. pp. 194–210 (2015)

104. Okubo, T., Ueda, K.: Peer-to-peer contents delivery system considering network distance. In: Network Operations and Management Symposium (APNOMS), 2011 13th Asia-Pacific. pp. 1–4. IEEE (2011)
105. Passarella, A.: A survey on content-centric technologies for the current internet: CDn and P2P solutions. Computer Communications 35(1), 1–32 (2012)
106. Poenaru, A., Istrate, R., Pop, F.: Aft: Adaptive and fault tolerant peer-to-peer overlaya user-centric solution for data sharing. Future Generation Computer Systems 80, 583–595 (2018)
107. Qamar, M., Malik, M., Batool, S., Mehmood, S., Malik, A.W., Rahman, A.: Centralized to Decentralized Social Networks: Factors that Matter, chap. 3, pp. 37–54. IGI Global (2016)
108. Qin, C., Yang, Z., Liu, H.: User interest modeling for P2P document sharing systems based on k-medoids clustering algorithm. In: Seventh International Joint Conference on Computational Sciences and Optimization (CSO). pp. 576–578. IEEE (2014)
109. Rabin, M.O.: Efficient dispersal of information for security, load balancing, and fault tolerance. J. ACM 36(2), 335–348 (Apr 1989), http://doi.acm.org/10.1145/62044.62050
110. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S.: A scalable content-addressable network. SIGCOMM Computer Communication Review 31(4), 161–172 (Aug 2001)
111. Ripeanu, M.: Peer-to-peer architecture case study: Gnutella network. In: Proceedings of the First International Conference on Peer-to-Peer Computing. pp. 99–100 (2001)
112. Risson, J., Moors, T.: Survey of research towards robust peer-to-peer networks: search methods. Computer Networks 50(17), 3485–3521 (2006)
113. Rosenfeld, A., Goldman, C.V., Kaminka, G.A., Kraus, S.: Phirst: A distributed architecture for P2P information retrieval. Information Systems 34(2), 290–303 (2009)
114. Rowstron, A., Druschel, P.: Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In: Middleware 2001. pp. 329–350. Springer (2001)
115. Schlosser, M., Sintek, M., Decker, S., Nejdl, W.: A scalable and ontology-based P2P infrastructure for semantic web services. In: Second International Conference on Peer-to-Peer Computing. pp. 104–111. IEEE (2002)
116. Schmidt, C., Parashar, M.: A peer-to-peer approach to web service discovery. World Wide Web 7(2), 211–229 (2004)
117. Schollmeier, R.: A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. In: Proceedings of the First International Conference on Peer-to-Peer Computing. pp. 101–102. P2P '01, IEEE Computer Society, Washington, DC, USA (2001)
118. Shah, V., de Veciana, G., Kesidis, G.: A stable approach for routing queries in unstructured p2p networks. IEEE/ACM Transactions on Networking 24(5), 3136–3147 (2016)
119. Sharan, A.: Exploiting semantic locality to improve peer-to-peer search mechanisms. Ph.D. thesis, Rochester Institute of Technology (2006)
120. Shen, X.J., Chang, Q., Gou, J.P., Mao, Q.R., Zha, Z.J., Lu, K.: Collaborative q-learning based routing control in unstructured P2P networks. In: MultiMedia Modeling. pp. 910–921. Springer (2016)
121. Shokouhi, M., Zobel, J., Tahaghoghi, S., Scholer, F.: Using query logs to establish vocabularies in distributed information retrieval. Information Processing and Management 43(1), 169180 (2007), http://research.microsoft.com/apps/pubs/default.aspx?id=80270
122. da Silva, P.M., Dias, J., Ricardo, M.: Mistrustful p2p: Deterministic privacy-preserving p2p file sharing model to hide user content interests in untrusted peer-to-peer networks. Computer Networks 120, 87–104 (2017)
123. Sripanidkulchai, K., Maggs, B., Zhang, H.: Efficient content location using interest-based locality in peer-to-peer systems. In: Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. vol. 3, pp. 2166–2176. IEEE (Mar 2003)

124. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. ACM SIGCOMM Computer Communication Review 31(4), 149–160 (2001)
125. Suel, T., Mathur, C., wen Wu, J., Zhang, J., Delis, A., Kharrazi, M., Long, X., Shanmugasundaram, K.: Odissea: A peer-to-peer architecture for scalable web search and information retrieval. In: International Workshop on the Web and Databases. pp. 67–72 (2003)
126. Sugiura, A., Etzioni, O.: Query routing for web search engines: Architecture and experiments. Computer Networks 33(1), 417–429 (2000)
127. Tang, C., Xu, Z., Mahalingam, M.: psearch: Information retrieval in structured overlays. ACM SIGCOMM Computer Communication Review 33(1), 89–94 (Jan 2003)
128. Tempich, C., Staab, S., Wranik, A.: Remindin': Semantic query routing in peer-to-peer networks based on social metaphors. In: Proceedings of the 13th International Conference on World Wide Web. pp. 640–649. WWW '04, ACM, New York, NY, USA (2004)
129. Tigelaar, A.S., Hiemstra, D., Trieschnigg, D.: Peer-to-peer information retrieval: An overview. ACM Transactions on Information Systems 30(2), 9:1–9:34 (May 2012)
130. Tirado, J.M., Higuero, D., Isaila, F., Carretero, J., Iamnitchi, A.: Affinity P2P: A self-organizing content-based locality-aware collaborative peer-to-peer network. Computer Networks 54(12), 2056–2070 (2010)
131. Tsoumakos, D., Roussopoulos, N.: Adaptive probabilistic search for peer-to-peer networks. In: Third International Conference on Peer-to-Peer Computing. pp. 102–109. IEEE (2003)
132. Tsoumakos, D., Roussopoulos, N.: Analysis and comparison of p2p search methods. In: Proceedings of the 1st international conference on Scalable information systems. p. 25. ACM (2006)
133. Ueda, K., Akase, J.i., Okubo, T.: Analysis of peer cluster layers selection criteria for P2P contents distribution systems. In: 15th Asia-Pacific Network Operations and Management Symposium (APNOMS). pp. 1–6 (2013)
134. Ueda, K., Okubo, T.: Peer-to-peer contents distribution system using multiple peer clusters. In: 14th Asia-Pacific Network Operations and Management Symposium (APNOMS). pp. 1–6 (2012)
135. Voulgaris, S., Kermarrec, A., Massoulié, L., van Oteen, M.: Exploiting semantic proximity in peer-to-peer content searching. In: Proceedings of the 10th IEEE International Workshop on Future Trends of Distributed Computing Systems. pp. 238–243. IEEE Computer Society, Washington, DC, USA (2004)
136. Wallach, D.S.: A survey of peer-to-peer security issues. In: Okada, M., Pierce, B.C., Scedrov, A., Tokuda, H., Yonezawa, A. (eds.) Software Security — Theories and Systems. pp. 42–57. Springer Berlin Heidelberg, Berlin, Heidelberg (2003)
137. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature 393(6684), 440–442 (1998)
138. Wu, K., Wu, C.: State-based search strategy in unstructured P2P. Future Generation Computer Systems 29(1), 381–386 (2013)
139. Wu, L.S., Akavipat, R., Menczer, F.: 6S: Distributing crawling and searching across web peers. In: Web Technologies, Applications, and Services. pp. 159–164 (2005)
140. Yan, F., Zhan, S.: A peer-to-peer approach with semantic locality to service discovery. In: Jin, H., Pan, Y., Xiao, N., Sun, J. (eds.) Grid and Cooperative Computing - GCC 2004, Lecture Notes in Computer Science, vol. 3251, pp. 831–834. Springer Berlin Heidelberg (2004)
141. Yang, B., Garcia-Molina, H.: Improving search in peer-to-peer networks. In: Proceedings 22nd International Conference on Distributed Computing Systems. pp. 5–14 (July 2002)
142. Yang, B., Garcia-Molina, H.: Improving search in peer-to-peer networks. In: 22nd International Conference on Distributed Computing Systems. pp. 5–14. IEEE (2002)
143. Yang, Y., Dunlap, R., Rexroad, M., Cooper, B.F.: Performance of full text search in structured and unstructured peer-to-peer systems. In: INFOCOM 2006. 25th IEEE International Conference on Computer Communications. IEEE Press (2006)

144. Yang, Z., Xing, Y., Chen, C., Xue, J., Dai, Y.: Understanding the performance of offline download in real P2P networks. Peer-to-Peer Networking and Applications 8(6), 992–1007 (2015)
145. Yu, W., Lin, X.: IRwr: incremental random walk with restart. In: Proceedings of the 36^{th} international ACM SIGIR conference on Research and development in information retrieval. pp. 1017–1020. ACM (2013)
146. Yu, Y.T., Gerla, M., Sanadidi, M.: Scalable vanet content routing using hierarchical bloom filters. Wireless Communications and Mobile Computing 15(6), 1001–1014 (2015)
147. Zeinalipour-Yazti, D., Kalogeraki, V., Gunopulos, D.: Information retrieval techniques for peer-to-peer networks. Computing in Science Engineering 6(4), 20–26 (2004)
148. Zeng, B., Wang, R.: A novel lookup and routing protocol based on can for structured p2p network. In: Computer Communication and the Internet (ICCCI), 2016 IEEE International Conference on. pp. 6–9. IEEE (2016)
149. Zhao, B.Y., Huang, L., Stribling, J., Rhea, S.C., Joseph, A.D., Kubiatowicz, J.D.: Tapestry: A resilient global-scale overlay for service deployment. Journal on Selected Areas in Communications 22(1), 41–53 (2004)
150. Zhu, Y., Wang, H., Hu, Y.: Integrating semantics-based access mechanisms with P2P file systems. In: Third International Conference on Peer-to-Peer Computing. pp. 118–125 (Sep 2003)
151. Zhu, Y., Hu, R., Fei, L.: A low latency resource location algorithm for unstructured P2P networks. In: International Conference on Computational Intelligence and Software Engineering. pp. 1–4. IEEE (2010)
152. Zhuge, H., Liu, J., Feng, L., Sun, X., He, C.: Query routing in a peer-to-peer semantic link network. Computational Intelligence 21(2), 197–216 (2005)