Distributed Semantic Overlay Networks

Christos Doulkeridis[1⋆], Akrivi Vlachou[1], Kjetil Nørvåg[2], and Michalis Vazirgiannis[1]

Abstract Semantic Overlay Networks (SONs) have been recently proposed as a way to organize content in peer-to-peer (P2P) networks. The main objective is to discover peers with similar content and then form thematically focused peer groups. Efficient content retrieval can be performed by having queries selectively forwarded only to relevant groups of peers to the query. As a result, less peers need to be contacted, in order to answer a query. In this context, the challenge is to generate SONs in a decentralized and distributed manner, as the centralized assembly of global information is not feasible. Different approaches for exploiting the generated SONs for content retrieval have been proposed in the literature, which are examined in this chapter, with a particular focus on SON interconnections for efficient search. Several applications, such as P2P document and image retrieval, can be deployed over generated SONs, motivating the need for distributed and truly scalable SON creation. Therefore, recently several research papers focus on SONs as stated in our comprehensive overview of related work in the field of semantic overlay networks. A classification of existing algorithms according to a set of qualitative criteria is also provided. In spite of the rich existing work in the field of SONs, several challenges have not been efficiently addressed yet, therefore, future promising research directions are pointed out and discussed at the end of this chapter.

[1]Department of Informatics, Athens University of Economics and Business (AUEB), 10434 Athens, Greece
[2]Department of Computer and Information Science, Norwegian University of Science and Technology (NTNU), 7491 Trondheim, Norway
{cdoulk,avlachou,mvazirg}@aueb.gr, Kjetil.Norvag@idi.ntnu.no

⋆ Corresponding author: Christos Doulkeridis, Athens University of Economics and Business, Department of Informatics, Patision 76, 10434 Athens, Greece, e-mail: cdoulk@aueb.gr, Tel: +302108203515.

1 Introduction

As data generation becomes increasingly distributed, either due to user-generated content (multimedia-images/documents) or because of application-specific needs (sensor networks, data streams, etc.), traditional centralized architectures fail to address the new challenges of contemporary data management. The massive amounts of distributed data, such as digital libraries and web accessible text databases, motivate researchers to work towards decentralized infrastructures for efficient data management and retrieval in highly distributed environments.

A promising solution for the design and deployment of distributed, global-scale applications is the exploitation of the peer-to-peer (P2P) paradigm. P2P has emerged as a prominent architecture for searching distributed data repositories, which reside on autonomous and independent sources. The overall challenge is for a large set of cooperative computers to support advanced search mechanisms over vast data collections, thus supporting a wide spectrum of applications.

Contrary to centralized systems and traditional client-server architectures, nodes that participate in a large P2P network store and share data in an autonomous manner. Such nodes can for example be information providers, which do not wish to expose their full data to a client. Therefore, it is challenging to provide efficient and scalable search, in a context of highly distributed content, without necessarily moving the actual contents away from the information providers. Then the problem of lack of global knowledge - in the sense that each peer is aware of only a small portion of the network topology and content - needs to be dealt with.

P2P systems can be distinguished into two main categories: *unstructured* and *structured*. In unstructured P2P, each peer maintains a limited number of connections (also called links) to other neighboring peers in the network. Searching in an unstructured P2P environment usually leads to either flooding queries in the network using a time-to-live (TTL) or query forwarding based on constructed routing indices that give a direction for the search. Examples of such unstructured P2P networks include Gnutella [17] and Freenet [6]. In structured P2P systems, a hash function is used in order to couple keys with objects. Then a distributed hash table (DHT) is used to route key-based queries efficiently to peers that hold the relevant objects. In this way, object access is guaranteed within a bounded number of hops. Examples of popular structured P2P systems are Chord [45], CAN [38], Pastry [40], Tapestry [54].

In general, searching in this context incurs high costs, in terms of consumed network bandwidth and latency. One of the important problems in P2P search is the high number of contacted peers that do not contribute to the final result set. Note that this is also the case for structured P2P networks, which are only efficient for exact search on the indexed key value. Semantic Overlay Networks (SONs) [9] have been proposed as an approach for organizing peers in thematic groups, so that queries can be selectively forwarded to only those peers having content within specific topics. In the case of unstructured P2P systems, SONs enable more efficient query routing to specific peer groups in a deliberate way, instead of blind forwarding. Although this problem is milder in structured P2P systems, as the search cost is logarithmic to the total number of peers, SONs are useful in this context as well, because they can increase the performance of the search [44].

The main topic of this chapter is an overview of existing algorithms and techniques for distributed semantic overlay network generation, in an unsupervised and decentralized manner. We first introduce the notion of semantic overlay networks and we describe the requirements for SON generation in Section 2. In order to illustrate the SON generation process, an approach for unsupervised, distributed and decentralized SON construction, named DESENT [12], is described as representative example in Section 3, which employs distributed clustering of peer contents, respecting the requirements imposed by the distributed nature of the environment [50]. Thereafter, in Section 4, we describe strategies for searching using SONs. We also show how the generated SONs can be exploited in a super-peer architecture, by having each super-peer responsible for a specific SON. Different application scenarios, including P2P web search and P2P image retrieval, are described in Section 5. Then, we provide a survey of existing SON algorithms and techniques in Section 6 and we present a taxonomy of these approaches using meaningful categorization criteria. Finally, in Section 7, we summarize the most important issues related to distributed SON generation and we outline the future research challenges.

2 Semantic Overlay Networks

In this section we describe the notion of semantic overlay networks in detail and we present a list of requirements for SON generation in a distributed P2P setting.

Semantic overlay networks have been originally proposed as an approach for organizing peers in thematic groups with similar contents, so that queries can be selectively forwarded to only those peers having content within specific topics. It should be mentioned that SONs do not necessarily imply use of semantics in the traditional sense (like ontologies), however this is the term first proposed in the literature [9] and it is used as such by all researchers.

An example of semantic overlay networks is illustrated in Figure 1. In the figure, peers are organized into three different SONs, according to their contents. These are: Sports & Activities, Music & Entertainment and Art & Culture. An important feature is that a peer may belong to more than one SON, as the peer can store data belonging to different thematic groups. For example, one peer has content relevant to both Sports & Activities and Music & Entertainment.

In the following, we present a set of desired goals for semantic overlay networks, in order to provide an assessment of SON quality. Then we proceed to identify and analyze a set of requirements for semantic overlay network generation.

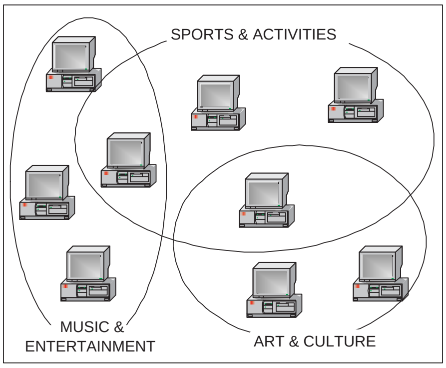


Fig. 1 The notion of semantic overlays in a P2P network.

2.1 Aims of SON Generation

The desired features of generated SONs are examined, in order to be able to evaluate and assess their quality. There exist two main measures that assist the assessment: *intra-SON similarity* and *inter-SON similarity*.

2.1.1 Intra-SON Similarity

An individual SON's quality is assessed independently of other SONs by the intra-SON similarity. This measure calculates the similarity of every pair of peers that belong to the SON. High values of intra-SON similarity show that peers contain relevant contents, hence the quality of the SON is high. Low values indicate that peers in the SON have dissimilar contents, therefore the quality of the SONs is not sufficiently high.

2.1.2 Inter-SON Similarity

When examining a set of generated SONs, inter-SON similarity is a suitable quality measure. Essentially, the similarity of every pair of SONs is calculated to determine how similar they are to each other. High values of inter-SON similarity mean that SONs describe common contents, so it is not easy to distinguish them. Low values show indicate that SONs are quite well-separated in terms of their contents, so any given query can be answered by only few SONs.

2.2 Requirements for SON Generation

Although several P2P research papers (see Section 6) adopt the use of semantic overlay networks, they also adopt a set of assumptions that more or less relax the basic constraints imposed by the P2P paradigm. In this section, we go a few steps back, as we identify with the benefit of hindsight from existing approaches the basic requirements for SON generation in a dynamic P2P environment: unsupervised algorithms, scalability, self-organization, autonomy and decentralization. We do not consider this list to be complete, we rather see it as a basic set of requirements that should be enforced, as they increase the value and benefit of any novel SON generation algorithm.

2.2.1 Unsupervised Algorithms

P2P networks in their initial, visionary form are systems characterized by lack of global knowledge. Instead, only local knowledge of content and topology can be safely assumed. However, several approaches make assumptions about the existence of background knowledge, in order to facilitate SON generation. A challenge is to organize the P2P network, assuming minimal pre-existing knowledge.

In principle, clustering algorithms are particularly suitable for SON generation, because they constitute an unsupervised approach. Apart from the input parameters that some clustering algorithms need to execute, distributed clustering of peers' contents assumes no further pre-existing knowledge. Nevertheless, several existing SON generation approaches rely on classification to group similar peers. The difference is that some background knowledge is assumed, usually in the form of a predefined taxonomy or as an already existing labeling scheme. While this assumption sounds reasonable for certain applications, it cannot by any means be generalized and presented as suitable for any P2P system. All in all, a need for unsupervised algorithms and approaches is identified.

2.2.2 Scalability

A unique feature of P2P computing is the unlimited scalability that can be achieved by exploiting the aggregate capabilities of all participating peers. Semantic overlays are proposed as a mechanism that improves the efficiency of search, so any such approach should be scalable. Potential bottlenecks in terms of communication costs (consumed bandwidth, latency, etc.) should be thoroughly studied. In the absence of sufficiently large testbeds, researchers use simulations to test the scalability of proposed systems.

Unstructured P2P networks, such as Gnutella, have problems related to scalability [39]. In particular, the time required to locate content in a large network can be extremely long, with high associated costs. Structured P2P systems solve this issue, by being able to find the answer to a query with logarithmic cost, however their feasibility is still questionable, especially because of the high maintenance cost of keeping the indexes updated. The dynamism of a P2P system, where peers may arbitrarily fail or join the network, poses another threat against ensuring scalable solutions.

Load-balancing is equally important, especially in the cases where the individual peer load may have an aggregate effect, i.e., increase with the size of the network. Any P2P system based on SONs should be able to scale well with the number of peers. Current P2P research focusing on SON generation should put scalability as number one requirement, as this need will become more evident in the future, where the urge for such viable, completely distributed systems is expected to increase.

2.2.3 Self-organization

Informally, the spontaneous activity towards organization of a system is described by the concept of self-organization. The basic mechanism for self-organization is dynamic topology adaptation, as a means to reorganize each peer's neighbors. In this way overlay networks are created on top of the initial P2P network topology. We stress here the essence of self-organization: there is no need for enforcing external observation and maintenance mechanisms. Additionally, it is not necessary for a system administrator to continuously set the values of system parameters or tune the system. Self-organization is one of the most challenging requirements in P2P systems and, at the same time, one of the most difficult to achieve.

2.2.4 Autonomy

Peer autonomy is an important concept in P2P networks, which is indirectly related to other issues like fault-tolerance. Peer autonomy means that each peer can be as independent as possible of the limitations imposed by the P2P protocol, concerning both its behavior and as well as its content. In particular, independence with respect to content means that each peer does not have to replicate its local data or provide explicit indices to its local data to other peers. Moreover, a peer should not be imposed to host indices to data that belongs to other peers. In this sense, unstructured P2P systems respect peer autonomy, in contrast with structured P2P systems. As a consequence, unstructured P2P systems are more resilient to failures, because in general, a peer failure makes only its local content unavailable, while in a DHT-based network, recovery mechanisms must be enforced for consistency. It is important that SON construction algorithms should also respect peer autonomy. However, peer autonomy comes with a cost: it is usually difficult to provide efficient search.

2.2.5 Decentralization

While distribution is inherently related to the P2P concept, the same does not always hold for decentralization. The most evident example of centralized P2P system was Napster, one of the first P2P systems to enable file sharing between participating computers. However, while the actual file exchange was performed in a P2P manner, the index was held in a centralized location. This is not an acceptable approach for dynamic P2P systems, since it presents a single point of failure.

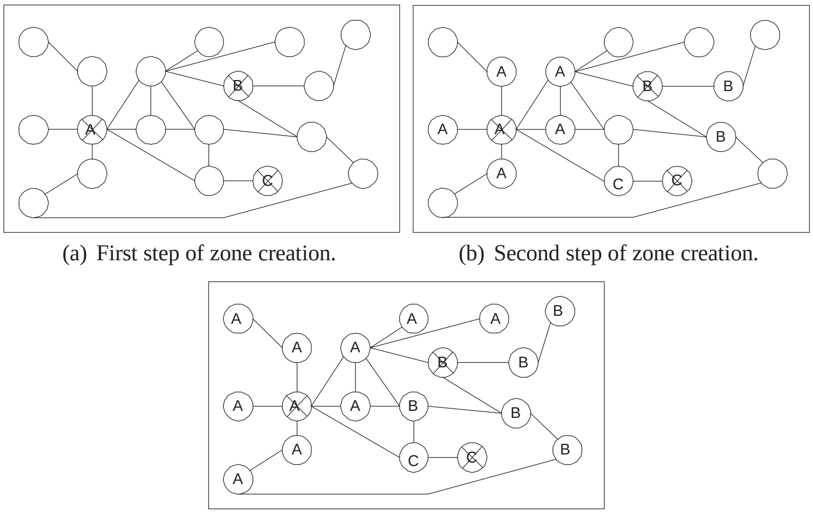
Learning from the shortcomings of such approaches, decentralization should also hold for SON generation, especially in large-scale networks. If operations are centralized, this endangers the completeness of SON generation, with obvious consequences to the correctness of the final overlays. Also, while for small networks a centralized solution may seem appropriate, due to the assembly of global knowledge, where better decisions can be made, however it usually presents problems when applied to large-scale networks. The main reason is communication bottlenecks that often result in non-applicable or infeasible approaches, or in other words algorithms that do not scale.

3 Distributed Creation of Semantic Overlay Networks

In this section we describe how the semantic overlays are created in a decentralized and distributed way using the DESENT approach as described in [12]. The approach is based on creating local topological groups of peers (called *zones*), forming clusters based on data stored on these peers, and then merging zones and clusters recursively until global zones and clusters are obtained. In this approach, clusters and semantic overlays are equivalent, and for the rest of this section we use these terms interchangeably.

3.1 The DESENT Approach to Decentralized and Distributed SON Creation

Having as a starting point the initial unstructured P2P network, some *initiator* peers are selected in a pseudo-random way (*initiator selection phase*). Initiators create local topological zones over their neighboring peers (*zone creation phase*). Then each initiator collects the cluster descriptions of all peers in its zone, and executes a clustering algorithm in order to create new clusters that span the entire zone (*zone clustering phase*). Since the clusters of two (or more) peers may be merged into a new cluster, this implies that these peers become members of a SON, and the SON's contents are now represented by a new cluster description. In the subsequent steps, the initiators form the current (unstructured) P2P network, thus playing the role of peers in the initial setup. Therefore, the process described above runs on the



(a) First step of zone creation.

(b) Second step of zone creation.

(c) Third step of zone creation.

Fig. 2 Step-wise zone creation given the three initiators A, B, and C.

initiators, thus new initiators are selected, that create zones and cluster zone contents in a completely similar way as shown above. Hence, zones and clusters are merged recursively until global clusters are obtained.

In order to be able to create zones of approximately equal size throughout the network (S_Z peers in each zone, where a typical zone size is in the order of $S_Z = 50$ peers), a subset of the peers are designated the role of zone initiators that can perform the zone creation process and subsequently initiate and control the clustering process within each zone. The process of choosing initiators is completely distributed, and essentially based on each peer assigning itself the role or not, using a function that is based on a combination of identifier and time. The result is that approximately one out of each S_Z peer assign themselves the role, and these peers are uniformly distributed in the network. One important feature of the initiator selection algorithm is that we obtain different initiators each time the algorithm is run. This tackles the problem of being stuck with faulty initiators as well as reducing the problem of permanent cheaters.

After a peer P_i has discovered that it is an initiator, it uses a probe-based technique to create its zone. An example of zone creation is illustrated in Figure 2, and as is illustrated an initiator gradually extends its zone until it finds a peer already belonging to another zone. This zone creation algorithm has a low cost wrt. to number of messages (in the infrequent case of excessive zone sizes, the initiator can decide to partition its zone, thus sharing its load with other peers). When this algorithm terminates, each initiator has assembled a set of peers Z_i and their capabilities (in terms of resources they possess), each peer knows the initiator responsible for its zone, and each initiator knows the identities of its neighboring initiators. An interesting

Algorithm 1 Zone-wise clustering.

1. The initiator of each zone i sends probe messages to all peers P_j in Z_i.
2. When a peer P_j receives a probe message it sends its set of feature vectors $\{F\}$ to the initiator of the zone.
3. The initiator performs clustering on the received feature vectors. The result is a set of clusters represented by a new set of feature vectors $\{F\}$.
4. The initiator selects a representative peer R for each cluster.
5. The result kept at the initiator is a set of cluster descriptions (CDs), one for each cluster C_i.
6. Each of the representative peers are informed by the initiator about the assignment and receive a copy of the CDs of *all* clusters in the zone.
7. The representatives then inform peers on their cluster membership by sending them (C_i, F_i, R).

characteristic of this algorithm is that it ensures that all peers in the network are contacted, as long as they are connected to the network. This is essential, otherwise there may exist peers whose content is never retrieved.

Independently of the actual construction of SONs, local clustering is performed on each peer resulting in a set of clusters. Each cluster is represented by a *feature vector* F_i which is a vector of tuples, each tuple containing a feature (word) f_i and a weight w_i. In order to reduce both computational and communication cost, only the top-k features are kept in F_i. After the zones and their initiators have been determined, global clustering starts by collecting feature vectors from the peers (one feature vector for each cluster on a peer) and creating clusters based on these feature vectors using Algorithm 1. In order to limit the computations that have to be performed in later stages at other peers, when clusters from more than one peer have to be considered, the clustering should result in at most N_C^0 such basic clusters (N_C^0 is controlled by the clustering algorithm). The result of this process is illustrated in Figure 3 (note that a peer can belong to more than one cluster). Each cluster C_i is described by a cluster description (CD), which consists of the cluster identifier C_i, a feature vector F_i, the set of peers $\{P\}$ belonging to the cluster, and the representative R of the cluster (the purposes of a representative peer is in some sense similar to a super-peer, and is among other things used to represent a cluster at search time), i.e., $\mathrm{CD}_i = (C_i, F_i, \{P\}, R)$. For example, the CD of cluster C_2 in Figure 3 (assuming A_7 is the cluster representative) would be $\mathrm{CD}_2 = (C_2, F_2, \{A_5, A_7, A_8, A_9\}, A_7)$.

At this point, each initiator has identified the clusters in its zone. These clusters can be employed to reduce the cost and increase the quality of answers to queries involving the peers in one zone. However, in many cases peers in other zones are able to provide more relevant responses to queries. Thus, we need to create an overlay that can help in routing queries to clusters in remote zones. In order to achieve this, we recursively apply merging of zones to larger and larger super-zones, and at the same time merge clusters that are sufficiently similar into super-clusters: first a set of neighboring zones are combined to a super-zone, then neighboring super-zones are combined to a larger super-zone, etc. The result is illustrated in Figure 4 as a hierarchy of zones and initiators. Note that level-i initiators are a subset of the level-$(i-1)$ initiators.

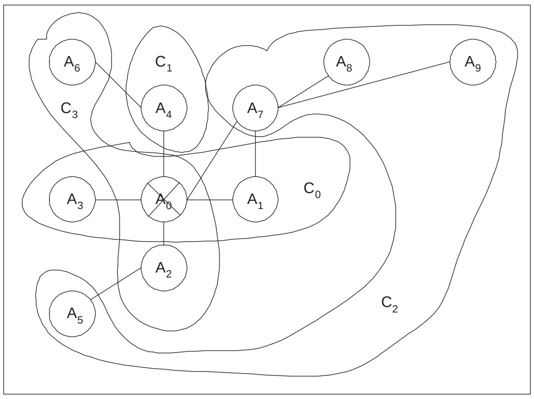


Fig. 3 Intra-zone clustering in zone A resulting in four clusters C_0, C_1, C_2, and C_3.

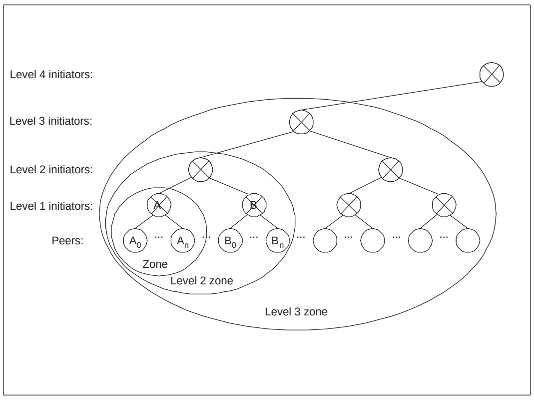


Fig. 4 Hierarchy of zones and initiators.

This creation of the inter-zone cluster overlay is performed as outlined in Algorithm 2, which is based on the overlay topology from the previous level of zone creation: since each initiator maintains knowledge about its neighboring zones (and their initiators), the zones essentially form a zone-to-zone network resembling the P2P network that was the starting point. In general, a level-i zone consists of a number of neighboring level-$(i-1)$ zones, on average $|SZ|$ in each (where SZ denotes a set of zones, and $|SZ|$ the number of zones in the set). This implies that $\frac{1}{|SZ|}$ of the level-$(i-1)$ initiators should be level-i initiators. This is achieved by using the same technique for initiator selection as described for the first level of zones, except that in this case only peers already chosen to be initiators at level-$(i-1)$ in the previous phase are eligible for this role.

Algorithm 2 runs iteratively (creating a new level at each iteration) until only one initiator is left, i.e., when an initiator has no neighbors. The only purpose of the final

Algorithm 2 Global clustering.

1. The level-i initiators create super-zones based on previous-level zones, and in this way also the level-i initiators become aware of their neighboring super-zones.
2. In a similar way to how feature vectors were collected during the basic clustering, the approximately $N_C|SZ|$ CDs created at the previous level are collected by the level-i initiator (where N_C denotes the number of clusters per initiator at the previous level).
3. Clustering is performed again and a set of super-clusters is generated.
4. Each of the newly formed super-clusters is represented by top-k features produced by merging the top-k feature vectors of the individual clusters.
5. A peer inside the super-cluster (not necessarily one of the representatives of the cluster) is chosen as representative for the super-cluster, resulting in a new set of CDs, $\text{CD}_i = (C_i, F_i, \{P\}, R)$, where the set of peers {P} contains the representatives of the clusters forming the base of the new super-cluster.
6. The CDs are communicated to the appropriate representatives.
7. The representatives of the merged clusters (the peers in $\{P\}$ in the new CDs) are informed about the merging by the super-cluster representative.

initiator that is produced by performing the algorithm is to decide the level of the final hierarchy, so it does not perform any clustering operations. The aim is to have in the end at the top level a number of initiators that is large enough to provide load-balancing and resilience to failures, but at the same time low enough to keep the cost of exchanging clustering information between them during the overlay creation to a manageable level. Thus, after Algorithm 2 has terminated, the top-level peer probes level-wise down the tree in order to find the number of peers at each level until it reaches level j with appropriate number min_F of peers. The level-j initiators are then informed about the decision and they are given the identifiers of the other initiators at that level, in order to send their CDs to them. Finally, all level-j initiators have knowledge about the clusters in zones covered by the other level-j initiators.

To summarize, the result of the zone- and cluster-creation process is two hierarchies: 1) a hierarchy of peers and 2) a hierarchy of clusters. Starting with individual peers at the bottom level, forming zones around the initiating peer which acts as a zone controller, neighboring zones recursively form super-zones (see Figure 4). On the top level are peers that effectively form a forest of trees, and where each peer has replicated the cluster information of the other initiators at that level.

Each peer is member of one or more clusters at the bottom level, and each cluster has one of its peers as representative. One or more clusters constitute a super-cluster, which again recursively form new super-clusters. At the top level a number of global clusters exist.

4 Searching in Semantic Overlay Networks

After semantic overlay networks have been successfully generated, the remaining issue is how to exploit them at query time, in order to improve the performance of searching. There exist two challenges in this context: 1) intra-SON search and 2) inter-SON search.

The former is about the way search is performed at peers that belong to a specific SON. For example, when a peer that belongs to a SON about rock music issues a query about ”Rolling Stones”, the query can be answered from this SON. However, the challenge is to find the appropriate peers to contact inside the SON. The latter refers to searches that cannot be served by the SON(s) that the querying peer belongs to. For instance, in the previous example, the peer issues a query about ”Shakespeare”, which clearly has to be forwarded to the SON responsible for literature. In this case, the problem is how to route the query to a particular SON, when a potential big list of SONs is available.

In the following, the most popular approach for SON-based search is presented. It consists of exploiting the links between peers with similar content for intra-SON search, and using links between peers with different content for inter-SON query routing. Unfortunately, this approach has several shortcomings, which motivate the need for a more efficient organization of SONs for searching. In this spirit, we demonstrate a super-peer architecture with a super-peer being responsible for a SON. The advantages of this approach are more efficient search relying in super-peer query routing mechanism.

4.1 Traditional SON-based Search

Most approaches that rely on semantic overlay networks generate two types of links between peers, in order to enable subsequent search. The first type of links is *short* links, which are used to connect peers with similar content and they are used inside a SON. The second type of links is *long* links and they enable search between different SONs. Representative approaches belonging to this category include [18, 23, 35].

Searching is then performed by routing queries over the appropriate links. For search within SONs short links are used, whereas search in different SONs is performed using long links. Usually, searching over short or long links takes the form of flooding, using a TTL value to limit the search cost.

Although the performance of search is increased compared to the basic search mechanism, one problem of this approach is the lack of a selective routing mechanism over the new links. In the case of large P2P networks, this can lead to increased searching costs.

Another problem arises when advanced query processing needs to be supported, instead of file sharing. Usually, in P2P file sharing, replicas of files exist in many peers in the network, and a request for a file can be satisfied by any one of these peers. In contrast, when the query is more complex, as for example finding the sum or average of data stored in peers, the query results will be exact only when all peers holding relevant data are contacted. As a consequence, there is a need for a more efficient mechanism that enables advanced query processing, by facilitating access to all peers that maintain data that contribute to the final result set.

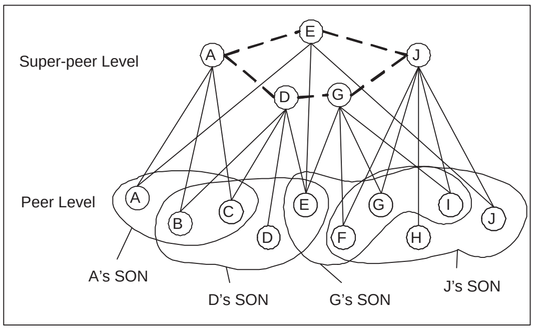


Fig. 5 SONs organized in a super-peer architecture.

4.2 Super-Peer Organization

As an improvement to the aforementioned problem, peers belonging to a SON are assigned to a super-peer. Thus, SONs are organized in a super-peer architecture, with each super-peer responsible for a SON. For an illustrative example see Figure 5. While the basic ideas are shortly presented here, the interested reader can find more details in [13].

Intra-SON search is performed by a peer contacting the super-peer responsible for the SON, and then the super-peer forwards the query to peers in the SON. This is performed in an efficient way, as it requires the minimum number of messages for query forwarding.

Inter-SON search is based on query routing at super-peer level. In the case that super-peers are organized in a particular super-peer topology, such as a hypercube (cf. [41]), super-peer routing is efficiently performed using the topology links. In the absence of a specific topology, there exists a naive search technique of flooding the super-peer network. However, even though the number of super-peers is typically at least an order of magnitude smaller than the number of participating peers, such a technique is not scalable. Therefore, a more efficient mechanism is to create routing indices [8] at super-peer level, in order to guide the search to appropriate super-peers.

As super-peer infrastructures [53] harness the merits of both centralized and distributed architectures, we make a design choice of a super-peer architecture for the P2P network interconnection. Moreover, the choice of the super-peer architecture is motivated and driven by our main requirement for scalability. Currently deployed super-peer networks, e.g. eMule (`www.emule-project.net`) and KaZaA (`www.kazaa.com`), support large number of users, demonstrating the appropriateness of super-peer architectures when scalability is required. In addition, peer autonomy is respected as the actual content remains on the owner peer, and only metadata is indexed and handled by the super-peer network.

5 Applications of Semantic Overlay Networks

In this section we present two application scenarios that exploit SON usage. First, we show how P2P document retrieval can be enhanced by adopting a SON-based approach (section 5.1). Thereafter, we present the design of a distributed search engine for multimedia content that also capitalizes on SONs (section 5.2).

5.1 P2P Web Search - Document Retrieval

The advent of the World Wide Web in combination with efficient search engines like Google and Yahoo! has made an enormous amount of information easily available for everybody. However, lately, several researchers have investigated the feasibility of providing web search facilities over a network of cooperative peers [22, 46, 32, 11, 4], contrary to the traditional model of centralized web indexing and searching. Some of the advantages of a completely distributed search engine are:

- Coverage and scalability: Current search engines only cover a small fraction of the documents on the Web, whereas a search engine consisting of thousands or millions of computers can scale better to achieve complete coverage.
- Freshness: Many web pages are rarely accessed by search engines, resulting in outdated search index contents,
- Avoiding information monopoly phenomena: A centralized search engine can control the flow of information, what is indexed, how it is presented to the user and – most importantly – the ranking. Censoring is also an issue as the search engine can filter out material that is considered controversial.
- Precision: one of the major concerns of current search engines is the precision of retrieval. A P2P search engine must adopt mechanisms that improve the precision of searching.
- Low cost: the cost of providing web search facilities employing the P2P paradigm is extremely low, as expensive data-centers and server farms are replaced by the commodity PCs of end-users.

A completely distributed search engine in its basic form consists of several individual peers that cooperate in order to provide the service collectively. Towards this goal, each peer uses part of its resources in order to build the necessary infrastructure for supporting web search. Therefore each peer:

- Collects the available web content, similar to a typical web search engine crawler.
- Indexes the assembled data, similar to inverted indexes.
- Publishes its index to the rest of the network, in order to make its data retrievable.
- Issues keyword-based queries for web document retrieval.

In the following, an application called SOWES[2] [11, 13] is presented, that supports efficient full-text search over an unstructured P2P network, by organizing peers into semantic overlay networks.

SOWES assumes a Gnutella-like unstructured P2P network of N_P peers and each peer stores a (potentially large) set of web documents. The documents stored at a peer may either belong to the peer (in case it is a web site) or may have been crawled from other sites on the web.

We now first describe how peers are organized into SONs, and then we present the searching mechanism for performing keyword-based queries.

5.1.1 SON Creation

Creation of semantic overlays in SOWES is a multi-phase distributed process, which capitalizes on DESENT (see Section 3). First zones of peers are created, then within each zone, SONs are created based on the data of peers. In the subsequent steps zones and clusters are merged recursively until global clusters are obtained. The main difference from DESENT is that the peer hierarchy is only used for SON creation and not for searching. When global SONs have been created, special purpose links are established between peers in SONs, in order to enable searching. Thus, the hierarchy of zones is only used during the SON creation process and finally the search is performed over a flat organization of SONs. In the end, the result is a set of SONs, each represented by a peer which functions similar to a super-peer.

We now describe in more detail the structure of SONs in SOWES; with particular emphasis on SONs interconnections and information retrieval aspects of SOWES. One important point is that when two individual SONs are merged into a new SON, the SON hierarchy is not maintained as in DESENT. Instead, links are created between peers in the two individual SONs ensuring sufficient connectivity, and then only the merged SON is used for further merging.

Similar to DESENT, each SON is a group of peers that is described by a cluster C_i of the peers' contents. Clusters are represented by cluster descriptors (CDs), one for each cluster C_i. In addition to the cluster identifier and feature vector, a random selection of N_r representative peers $\{R\}$ belonging to the cluster is part of the cluster description. The value of N_r is chosen high enough to minimize the probability of all of them disappearing, due to peer failures, during the lifetime of the CD, but low enough to avoid high communication cost when communicating the CDs. As an example, considering the peers in Figure 3, the CD of cluster C_2 would be $\mathrm{CD}_2 = (C_2, F_2, \{A_5, A_8\})$, assuming A_5 and A_8 are the randomly chosen peers.

In order to ensure the connectivity of the merged SONs, d links are used between the merged SONs. In this way, the probability that a SON becomes disconnected due to peer failure is eliminated. The d links between the peers of the two clusters are formed by iteratively selecting from each SON, the least connected peers, and then

[2] **S**emantic **O**verlays for **We**b **S**earching

connecting them. The algorithm ensures that there exists a path from each peer to any other peer within each SON.

As described so far, the resulting SONs are sufficiently connected internally. However, in order to support inter-cluster queries, the resulting SONs also need to be sufficiently connected. This is performed during the last step, when there is one zone and it is achieved by creating d connections from each representative peer to representatives of other SONs. This ensures connectivity at cluster representative level and consequently enables query routing among SONs.

Before describing how query routing is performed, we briefly review the connectivity of each peer Q:

- Q maintains L_d connections to other peers that belong in the same global SON, for each global SON that Q belongs to. These connections were created during the SON merging phase.
- Q maintains L_p connections to cluster representatives, for each global SON the peer belongs to.

In addition, cluster representatives of the global SONs maintain L_g connections to other cluster representatives. These connections are used to ensure inter-SON communication, in order to make any SON reachable from any peer and at the same time avoid its potential isolation from the rest of the SONs.

5.1.2 Keyword-based Search

In this section, we describe how keyword-based search is performed. We first describe in more detail the basic approach to searching, followed by techniques that reduce the cost of query routing. A query for web documents originates from a querying peer Q_P. In an unstructured P2P system, querying is performed by routing the query to appropriate peers and performing the query on each of these peers, and then returning matching results. In our context, processing the query is performed by first determining which clusters might contain relevant data (*inter-cluster routing*), followed by searching one or more of these clusters (*intra-cluster routing*).

Inter-cluster routing refers to query routing at cluster representative level, which aims to identify similar cluster descriptions to the query. In order to limit the amount of costly intra-cluster searching, the inter-cluster routing is performed in two steps.

In the first step, a search for appropriate clusters is performed. A number of techniques can be used to find these clusters. Potential solutions include *random jumps* or some *gossiping approach* that allow peers to become familiar with a small set of peers outside their cluster. However, both these techniques and their variants impose a query horizon, i.e., they cannot guarantee that remote peers are reached in very large networks. In order to overcome such limitations, we use the links created among cluster representatives to route queries. In the absence of more sophisticated mechanisms, this routing takes the form of flooding. In this way, we can guarantee that the query contacts all clusters (i.e., their cluster representatives), thus enabling access even to the most distanced peers. Those cluster representatives reached in this way having a similarity to the query that is larger than a certain threshold, return their CD to the originating peer Q_P.

In the second step, Q_P determines which clusters are most appropriate (based on the results of step 1), and forwards the query to these for intra-cluster searching. In a real system the number of clusters to search is determined by the number of results returned, so that the number of clusters searched can be limited.

Intra-cluster routing refers to query routing within a cluster. Routing always starts from a cluster representative (which has been found during inter-cluster routing as described above), and it is performed as flooding starting from the representative through its L_d connections. Each peer that received the query executes it locally and if it has matching results these are returned to Q_P.

5.2 P2P Image Retrieval

The widespread use of digital image equipment enables end-users to capture and edit their own image content, sometimes of high intellectual or commercial value. The centralized character of Web search raises issues regarding royalties, in the case of protected content, censorship, and to some degree information monopoly. Moreover current tools for image retrieval are limited in query expressiveness and lack semantic capabilities. Image content is only partially covered by web search engines, although it is evident that there is a tremendous wealth of digital images available on computers or other devices around the world. This is probably due to the fact that image content induces further complicated problems regarding the search: current centralized image search facilities do not sufficiently support metadata management, semantics, advanced media querying etc.

In this section, we present a SON-based architecture that adopts the P2P paradigm for indexing, searching and ranking of image content. The ultimate goal of our architecture is to provide a search mechanism for image content relying on image features. The overall system is described in [51] and the algorithms for query processing are introduced in [14]. The application described in this section is a scalable, decentralized and distributed infrastructure for building a search engine for image content capitalizing on P2P technology.

5.2.1 Architecture

The proposed architecture utilizes a P2P infrastructure for supporting the deployment of a scalable search engine for multimedia content, addressing future user needs and search requirements. A high-level abstraction of the architecture, showing the different participating entities, their interconnections and functionality is shown in Figure 6.

Collaborating entities in the search engine consist of: 1) *content providers and requestors* and 2) *information brokers*.

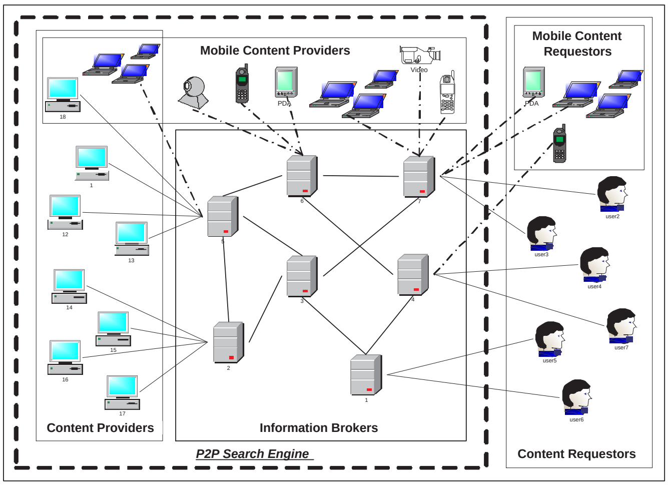


Fig. 6 Architecture of P2P search engine for image content.

Content providers are entities that produce or store images that they would like to publish. *Content requestors* are not necessarily contributing content to the search engine. However their role is important, as they represent the users of the search engine. These roles are not mutually exclusive. Content requestors enjoy a rich repertoire of query and searching facilities and they are provided access to thousands or millions of independent and undiscovered multimedia sources (content providers). Besides stationary content providers, mobile content providers which are able to dynamically capture content, also like to exchange image data and make it widely available.

Information brokers consist of more powerful and less volatile peers in the network. They realize a decentralized indexing service. In addition to the basic form of metadata that is generated by the content providers, information brokers may employ more sophisticated (and thus demanding) algorithms that could not be executed in lightweight peers, due to lack or processing power or lack of more widespread knowledge on the rest of the multimedia content in the network.

Obviously, the afore-described architecture can be realized by means of a super-peer network. Content providers are peers that want to make their content searchable and also be able to search other peers' content, acting as content requestors. On the other hand, information brokers are super-peers that form the backbone of the search engine. In what follows, we describe how SONs can improve the plain super-peer architecture, in terms of search performance.

5.2.2 Peer Organization

One of the most important factors that influences the performance of P2P systems is the underlying data distribution to peers (and super-peers). In the general case, when a peer joins the network, it connects to a super-peer, usually in a random way. However, data on peers is usually clustered into a few thematic areas that reflect the users interests. It is therefore necessary to detect peers with similar content that belong to the same community. Additionally, this should occur in a self-organizing way, without explicit external intervention. Then, queries can be directed to specific peers only, thus improving query processing performance.

Semantic overlay networks can improve the basic architecture of the distributed search engine for image content. SONs are created in order to form communities that store similar content. After SONs have been formed, a super-peer become responsible for one or more SONs. This is achieved by reassigning peers to super-peers, in such a way that the assignment corresponds to SON membership. The result is that a super-peer indexes peers that store similar content. Consequently, queries can be answered by being directed to a limited set of super-peers, which are responsible for content relevant to the query.

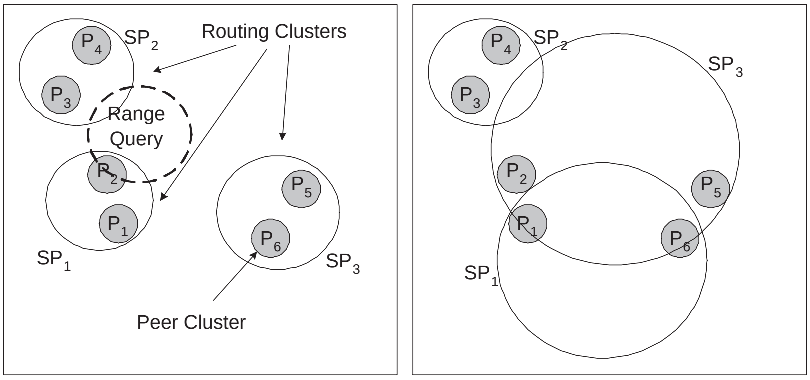
5.2.3 Similarity search

The SON-based super-peer architecture is a good starting point for constructing an efficient search mechanism. The aim is to support queries like ”retrieve all images similar to a given an image” or ”retrieve the *k* most similar images to a given one”.

Similarity search over image content usually involves having images represented by objects in a high dimensional data space. Features of this data space may include both text-based features (such as key words, annotations) and visual features (such as color, texture, shape, faces). This feature extraction is a semi-automatic process and it takes place on each peer, before joining the search engine. Then similarity search is performed by computing the similarity or distance of high dimensional objects, provided that there exists a commonly accepted similarity or distance function.

SIMPEER [14] is a super-peer system that supports similarity search over high-dimensional data. This system can realize the search mechanism required on top of the architecture described above. After presenting its functionality in short, we show the difference in query processing performance induced by a SON-mechanism.

SIMPEER relies on a three-level clustering scheme and supports efficient P2P similarity search in metric spaces. Given a super-peer network, each peer connects to a super-peer and maintains its own data, represented in a high dimensional space. In a construction phase, each peer applies a clustering algorithm on its local data. Thereafter, each super-peer gathers the clusters of its associated peers and applies on them a clustering algorithm resulting in a new cluster set that describe the data indexed by this super-peer. These clusters are broadcasted at the super-peer network, in order to form *routing clusters* at super-peers.



(a) SON-based routing clusters. (b) Random routing clusters.

Fig. 7 Super-peer routing clusters and range query.

Figure 7(a) depicts the routing information stored at each super-peer, namely the routing clusters of all other super-peers. The cluster information of peers is stored only at the corresponding super-peer of each peer. At query time, each super-peer decides where to forward a query, based on its routing clusters. Assume a range query, initiated at a super-peer SP_q. First, SP_q examines its routing clusters to find to which of its neighboring super-peers the query should be forwarded to. Each recipient super-peer SP_r checks whether its local peers can provide any results, by inspecting their clusters. If the query overlaps with some clusters, SP_r contacts only the peers responsible for these clusters. Otherwise, SP_r simply forwards the query to its neighboring super-peers, using its routing clusters.

Figure 7 depicts the effects of SONs for our search engine using SIMPEER. In Figure 7(a) the peer clusters assigned to a routing cluster are similar, leading to high intra-SON similarity. The inter-SON similarity, i.e. the overlap between routing clusters of different super-peers, is low. In this case SIMPEER has been shown to work well, meaning that only few super-peers and peers that can provide relevant results are queried, resulting in reduced network traffic and response time. However, for the case depicted in Figure 7(b), each peer cluster is assigned randomly to a super-peer, resulting in routing clusters that are not well-separated. Thus, the routing ability of the routing clusters is reduced, while the network traffic and the number of contacted super-peers are increased.

6 Related Work

Semantic Overlay Networks (SONs), have been proposed as an approach for semantically organizing peers, so that queries can be forwarded to only those peers containing documents within specific topics. In the seminal paper [9], SONs are presented as thematic focused groups of peers, which share common interests. In this respect, a P2P network is organized into SONs, in order to enable efficient routing of queries only to relevant peers. The advantage of this approach is that it reduces the routing cost, namely it reduces the flooding cost in the case of unstructured P2P networks and decreases the number of peers that need to be contacted in the case of structured P2P systems. In principle, SONs also enhance the quality of results, an important issue in the case of distributed information retrieval tasks like P2P web search.

In the following, we review existing algorithms and techniques for semantic overlay network generation in P2P systems. A taxonomy of semantic overlay networks is presented in Table 1, by first distinguishing them based on the underlying P2P architecture, namely *unstructured* and *structured*. Then, within each architecture, three basic methods for SON generation are identified: *clustering*, *classification* and *gossip-based*.

	Unstructured	**Structured**
Clustering	DESENT [12] Distributed K-means [15] Topic-segmented overlays [3] SSW [23] Super-peers [13] Klampanos *et al.* [19] Semantic Peer [26]	pSearch [47] WonGoo [31] SWOP [18]
Classification	Content-based Overlays [27] Fairness Index [49] iClusterDL [37] iCluster [35]	HSPIR [25] Content-based Overlays [27]
Gossiping	REMINDIN' [48] Associative overlays [7] Shortcuts [44] INGA [29] Acquaintances [5] Metric Social Networks [42] SON reformulation [20] Epidemic protocol [52] Chatty Web [1] Language Models [24]	p2pDating [34] GridVine [2]

Table 1 Taxonomy of semantic overlay network generation algorithms.

6.1 Unstructured P2P

6.1.1 Clustering

Although several papers describe how to use SON-like structures, little work exists on the issue of to actually create SONs in an unsupervised, decentralized and distributed way in unstructured networks. One important difficulty arises when there is a lack of background knowledge about peer contents. In the following, SON construction methods that use *unsupervised learning* are presented, with most of these approaches relying on *clustering algorithms*.

A distributed (but not decentralized) P2P clustering algorithm is presented in [15], so that for a given query, the querying peer can identify the most promising clusters and route the query precisely to peers containing relevant documents.

In [3], *topic-segmented overlays* are proposed, which partition peers into groups of similar topics. The approach is based on assembling peer representations at one site and perform a clustering that is then communicated to all peers. Cluster updates are treated similarly; the site determines any changes of the global cluster descriptions and informs other peers on demand.

Klampanos et al [19] propose to cluster peer contents, in order to generate a cluster-based architecture for P2P information retrieval. Individual peer documents are clustered using a hierarchical clustering algorithm and document clusters are evaluated using two metrics: 1) the average standard deviation of term frequency among the documents of each cluster and 2) the number of peer documents within a specific topic, which shows the expertise of a peer. Then a one-pass peer clustering algorithm is used to identify *content-aware groups*, i.e. peer clusters. However this algorithm is not distributed, assuming all knowledge of peer clusters is available at one location.

Semantic small world (SSW) [23] aims at small world network construction by semantic clustering of peer contents. Each peer is represented by the centroid of its largest cluster in the semantic space. This forms the peer's semantic label. Furthermore, each peer maintains short and long range contacts, where short range contacts refer to neighboring peer clusters, while long range contacts refer to peers with data at randomly chosen points in the semantic space.

In [13], SONs are generated using hierarchical clustering of peer contents. After SON construction, each SON is assigned to a super-peer, resulting in a super-peer organization of SONs. This leads to performance improvements, both in terms of retrieval quality and associated search costs.

Recently, a measure for cluster cohesion, called *clustering efficiency*, has been proposed in [36], as a way to evaluate SON quality. In contrast to existing measures, clustering efficiency takes into account the neighborhood of a peer, not only its direct neighbors. Maintenance of neighboring links is based on a periodic rewiring procedure, which triggers the process of discovering similar peers.

Another approach to improve on some of the problems of Gnutella-like systems, is to use a super-peer architecture where a number of peers/clients are connected to a super-peer, which 1) indexes the data stored on its peers, as well as 2) communicates with other super-peers so that queries that can not be satisfied from the local super-peer can be forwarded to other super-peers. Super-peer approaches benefit from the advantages of both the centralized and the distributed paradigm. They combine the efficiency of centralized search with the intrinsic features of P2P. An interesting study of super-peer networks is presented by Yang and Garcia-Molina [53]. The authors discuss design issues concerning super-peer networks and come up with a set of useful rules of thumb. Edutella [33] is another super-peer approach, based on a hypercube topology [41], for routing queries. Super-peers maintain indices of peer contents and routing indices, and searching is practically achieved through routing at super-peer level. A super-peer architecture can also be used to realize a hierarchical summary index as described in [43]. In [28], clustering policies are proposed to generate semantic clusters in super-peer networks. Particular emphasis is put on managing heterogeneous data schemes.

6.1.2 Classification

One of the problems of unsupervised techniques is that it is generally difficult to identify coherent clusters, without exploiting any background knowledge. As a solution to this, several papers have advocated the use of background knowledge, in order to generate groups of peers with similar content. These methods belong to the category of *supervised learning* and they usually employ *classification algorithms* or assume the existence of a common ontology among peers.

In [49], a P2P architecture where nodes are logically organized into a fixed number of clusters is presented. The main focus is on fairness with respect to the load of individual nodes. For this reason, the *fairness index* is proposed, as a metric for inter-cluster load balancing. Furthermore, intra-cluster load balancing is considered, in order to achieve load balancing among peers that belong to the same cluster.

In [35], the authors present *iCluster*, a system where peers become members of SONs by creating and maintaining two types of links: *short-range* and *long-range* links. Short range links connect peers inside a SON, while long range links are used to interconnect SONs and assist searching. The actual SON creation is based on a periodic rewiring procedure, with each peer trying find other similar peers. An application of this approach in a digital library context is the *iClusterDL* system [37]. Notice that according to the authors SONs can be generated either through document clustering or classification, however we choose to record this work as classification mainly due to the way experiments were performed.

In [26], the authors address the issue of SON creation in a Peer Data Management System (PDMS) where peers have different schemas and they are connected through schema mappings. Each peer is represented by a set of concepts. Each concept will be associated to at most one SON. Heterogeneity is solved using the WordNet as background thesaurus. The authors define a distance function among sets of concepts and they use the BUBBLE framework [16] for clustering. At peer join time, neighbor selection can be performed in two ways: range selection (select the peers in the SON with distance below a threshold) and k-NN selection (the k closest peers semantically). Especially the k-NN approach, requires setting a value for the radius, which requires some non-negligible communication and synchronization with the joining peer. Updates seem to flood the network and their propagation does not have a stopping mechanism.

6.1.3 Gossip-based

Several papers related to SON construction rely on gossiping protocols for distributing information in the P2P network. Once information about peers' contents is spread in the network, a peer can establish links to relevant peers. The advantage of gossip-based approaches is that they are completely distributed and support self-organization. However, one of their disadvantages is that they can provide no guarantees that relevant peers will be acquainted in finite time, especially in the case of really large and dynamic P2P networks. Obviously, depending on the application, gossip-based approaches can provide an efficient solution for SON generation and subsequent searching.

Associative overlays, proposed in [7], are similar to SONs. The underlying concept is that peers which have previously satisfied queries are more likely candidates for answering a current query. Hence, such peers are organized into overlay networks. The approach has advantages over blind search in unstructured P2P networks, especially for searches for rare items.

A similar notion has been recently proposed. In *Metric Social Networks* [42], peers establish links based on query histories, thus forming overlay networks for improving the efficiency of similarity search. The main idea is to exploit the concept of social networks, with peers creating relationships to other peers according to the similarity of their data. Peer links are generated based on query results, trying to identify peers that contributed significantly to the results of a query.

Cholvi *et al.* [5] propose the use of *acquaintances* as an extension to Gnutella-like networks to improve searching. Peers with similar contents are linked together, so that searches for a particular topic can be routed to more relevant peers in less time. Semantic communities of peers are created in this way. The basic intuition of this approach is that efficient searching is achieved because requests that initiate from a particular community can be fulfilled within the community with high probability.

In [44], peers that share similar interests create *shortcuts* to each other, thus enriching the original overlay connections. The use of interest-based shortcuts enhances the basic search mechanism employed by Gnutella. Essentially, the creation of interest-based groups of peers is quite similar to the concept of semantic overlay networks.

The use of shortcuts has also been employed for generating semantic social overlay networks [29]. In the system called INGA, peers create and maintain shortcuts to other peers in a lazy manner, by examining past queries both issued by the peer itself and routed through the peer. Shortcuts are maintained at four different level serving different purposes: 1) at the content provider layer, shortcuts are created to remote peers which have successfully answered a query, 2) at the recommender layer, information is maintained about remote peers who have issued a query, 3) at the bootstrapping layer, shortcuts to well connected remote peers are kept, and 4) at the network layer, connections are maintained to peers of the underlying network topology. The authors study the comparative performance of INGA to the approach presented in [44] and their results show that INGA performs better.

Yet another approach where peers collect and maintain information about other peers that answered successfully past queries is presented in [48]. The authors study query evaluation of RDF data distributed over data repositories. Similarly to previous works, the objective of keeping information of successful past query evaluation is to enable future peer selection. This approach has been later extended, in order to support ranking of peers [30].

Aberer *et al.* [1] present a similar gossip-based technique for creation of semantic links among peers. The main focus of the paper is on producing global semantics from local interactions among peers.

In [24], statistical language models on peers are used to describe peer contents. Then SONs are created by having peers periodically exchange information about their nearest neighbors in the semantic space. Through random meetings with other peers, each peer maintains locally a set of references about interesting peers, which will be used at query time.

Voulgaris *et al.* [52] propose an epidemic protocol for SON construction. This is essentially a gossiping approach that enables peers to become familiar with other semantically related peers. In this way peer clustering is performed using a semantic proximity function that quantifies the semantic similarity of peers.

In [20], the authors study the problem of keeping SONs updated, without having to reconstruct the overlay network. The approach is based on having a cluster representative for each cluster, which assembles relocation requests and serves them in a second phase. The cluster reformulation problem is modeled in a novel way as a game, where peers decide which clusters to join based on potential gains in the recall of their future queries.

6.2 Structured P2P

6.2.1 Clustering

In the context of structured P2P systems, several research papers have dealt with the issue of improving the basic search provided by the underlying DHT. In general, two are the main goals that should be achieved. First, in document retrieval applications, support for semantic retrieval is required, in contrast to existing exact matching of keywords to documents. Semantic overlay networks are useful in this direction, as peers are organized based on the semantics of their content. Second, improvements to the performance of the underlying structured P2P protocol can be achieved, using SON-based techniques. The objective is to reduce the search cost and the increase the precision of the search, by contacting only carefully selected peers.

Tang *et al.* [47] propose pSearch, a system for SON creation on top of structured P2P networks. The approach is based on the use of LSI to compute the (multidimensional) semantic key that can provide the coordinates for a document in a CAN network, and then use a nearest-neighbor search technique in order to find appropriate documents. Thus the indices of semantically related documents are assigned to the same or neighboring peers in the overlay.

Improvements to the approach are presented in WonGoo [31]. Mainly the approach tries to improve on the costly update process of pSearch. This is achieved by computing LSI of terms not documents, so as terms in a corpus are not significantly affected when the corpus changes, the LSI of terms do not need to be updated. In terms of the actual SON generation, WonGoo uses similar techniques with pSearch.

However, some of the inherent problems of LSI, like processing cost and choosing number of dimensions, make it difficult to employ this technique in a large-scale dynamic document repository. Furthermore, both [25] and [47] assume a central LSI computation, which presumes that all documents or a sample of reasonable size must be assembled at a central location, which is infeasible for a large P2P system, especially when acquisition of global knowledge is impossible.

In [18], the SWOP protocol for construction of small-world P2P overlays has been proposed. Similar to other papers, peers maintain *cluster links* and *long links* to neighboring and distant nodes respectively. In this way, small-world overlay network can improve the performance of the underlying structured P2P, in this case Chord. Furthermore, exploiting the nice properties of small world network, the authors propose an effective object replication algorithm, suitable for handling dynamic query workloads for popular objects, such as the flash crowd scenario.

6.2.2 Classification

There are only few papers that use classification techniques for semantic overlay network creation over structured P2P systems.

Liu *et al.* [25] propose HSPIR, a hierarchical SON based on CAN [38] and Range Addressable Network [21]. Support for semantics is achieved by using Latent Semantic Indexing (LSI) [10]. A document is represented using the vector space model and LSI and it is classified using support vector machines. A query is classified in order to match a topic and then routed to the appropriate peer responsible for this topic using CAN. Then this peer forwards the query to other peers inside the SON in a hierarchical way, exploiting the range addressable network.

In [27], the authors present a system that exploits existing classifications of peers in taxonomies, in order to build overlays with peers based on taxonomy information. This enables the creation of *Content-based Overlay Networks (CONs)*, which is essentially a content-based grouping of peers. One advantage of CONs over traditional DHT-based systems is that queries that retrieve only few results can be broadened exploiting the information stored in the taxonomy.

6.2.3 Gossip-based

A similar approach to [5, 44] has been described in [34]. Peers perform random meetings with other peers at regular intervals, thereby progressively establishing connections to *friendly* peers. An advantage of this approach is that it respects peer autonomy, as each peer may independently decide which connections to create and which to avoid. Another nice feature of this family of techniques is adaptivity to changes in the peer's underlying content and interests. The authors present this SON generation method as an improvement of P2P web search over a structured P2P infrastructure [32].

A different notion of SONs, as outlined in [2], is related to schema mappings and peers that are logically interconnected through schema mappings. The *GridVine* system is built on top of a structured P2P network, called P-Grid. Through local schema translations and semantic gossiping, global SONs are created.

6.3 Evaluation

In this section, we present a meaningful evaluation of the aforementioned approaches. The aim is to identify their comparative advantages and disadvantages based on the nice properties that are supported by each approach. A summary of this comparison is presented in Table 2.

A set of quality indices is employed, in order to perform the evaluation. A SON generation method is evaluated based on its adherence to the following properties: distributed, decentralized, unsupervised, scalable, delf-organizing, autonomy. For a more general discussion regarding the requirements for SON generation we refer to [50].

The rationale for classifying a SON generation algorithm is briefly explained in the following. An algorithm is distributed when it is executed at several sites, instead of a single one. Most SON generation algorithms fulfill this criterion, with few exceptions, for example the approach in [19] describes a peer clustering algorithm that is not distributed. Decentralization characterizes algorithms that do not require a centralized location for assembling data or processing algorithms. For instance, pSearch [47] requires the computation of LSI in a centralized manner, therefore it is not decentralized. The distinction between supervised and unsupervised approaches is easy, as it is based on the existence of background knowledge or the use of supervised learning techniques. Scalability is an important requirement and it is not always supported by SON generation methods. An approach is deemed scalable if it can scale up to hundreds of thousands of peers, as opposed to some hundreds or thousands of peers only. For such large-scale networks, it is unclear whether gossip-based approaches can produce results of acceptable quality, as they are usually evaluated at much smaller scale. Self-organization is a necessary property for P2P systems, and approaches that work in a bottom-up manner, starting at peer level and converging at some fixed point without external intervention, usually satisfy this requirement. Peer autonomy is the typical case for SON creation methods in unstructured P2P networks, however it is often violated in structured P2P systems, when peers are assigned specific parts of the global index.

		Distributed	Decentralized	Unsupervised	Scalable	Self-Organizing	Autonomy
Unstructured P2P	DESENT [12]	√	√	√	√	√	√
	Distributed K-means [15]	√		√	√	√	
	SSW [23]	√	√	√	√	√	
	Klampanos *et al.* [19]			√	√	√	√
	Topic-segmented overlays [3]			√		√	√
	Fairness Index [49]	√	√		√	√	√
	Super-peers [13]	√	√	√	√	√	√
	iClusterDL [37]	√	√	√		√	√
	iCluster [35]	√	√	√		√	√
	REMINDIN’ [48]	√	√	√		√	√
	Associative overlays [7]	√	√	√		√	√
	Shortcuts [44]	√	√	√		√	√
	INGA [29]	√	√	√		√	√
	Acquaintances [5]	√	√	√		√	√
	Metric Social Networks [42]	√	√	√		√	√
	SON reformulation [20]	√	√	√		√	√
	Epidemic protocol [52]	√	√	√		√	√
	Chatty Web [1]	√	√	√		√	√
	Language Models [24]	√	√	√		√	√
	Semantic Peer [26]	√	√	√		√	√
Structured P2P	pSearch [47]	√			√	√	
	HSPIR [25]	√			√	√	
	WonGoo [31]	√			√	√	
	SWOP [18]	√	√	√	√	√	
	p2pDating [34]	√	√	√		√	√
	Content-based Overlays [27]	√	√		√		
	GridVine [2]	√	√		√		

Table 2 Evaluation of existing semantic overlay network techniques.

7 Summary and Future Trends

In this chapter, distributed semantic overlay networks for peer-to-peer systems have been presented. Having described the requirements for overlay construction, a semantic overlay network generation algorithm with salient features has been presented. Furthermore, searching over semantic overlay networks has been discussed, demonstrating the advantages in terms of improved performance. A number of applications can be deployed exploiting semantic overlay networks and two such examples have been presented, namely P2P web search and P2P image retrieval. Finally, a comprehensive overview of the research related to semantic overlay networks has been conducted.

Future research related to semantic overlay networks looks promising. One research area that has attracted interest lately is social networks. Semantic overlay networks can help to identify social communities in the case that such explicit information is not available. In addition, more advanced tasks like social network mining can be performed using techniques and experiences from semantic overlays.

Another interesting research direction is peer ranking in the context of semantic overlays. Most approaches so far consider peers inside a SON as equal, however there exist several applications that could exploit ranking such peers according to specific criteria. For example, in the case that an application needs to return a few results fast to the user, contacting the highest ranked peers instead of all, would result in savings in response time. Such an application is top-k retrieval, which is quite common in today's web search.

Personalization is an interesting topic that can be studied in the context of semantic overlay networks. In scenarios where users have different preferences or when objective criteria either do not exist or are not helpful, it is not possible to establish commonly accepted semantic overlays. Thus it is interesting to study semantic overlay generation and maintenance in uncertain environments.

In spite of the rich existing work in the field of SONs, several challenges have not been efficiently addressed yet, therefore, distributed SON management is deemed as a future promising research direction.

References

1. Aberer, K., Cudré-Mauroux, P., Hauswirth, M.: The chatty web: emergent semantics through gossiping. In: Procedings of 12th International Conference on World Wide Web (WWW), pp. 197 – 206 (2003)
2. Aberer, K., Cudré-Mauroux, P., Hauswirth, M., Pelt, T.V.: GridVine: Building internet-scale semantic overlay networks. In: Proceedings of International Conference on Semantic Web (ISWC), pp. 107–121
3. Bawa, M., Manku, G.S., Raghavan, P.: SETS: Search enhanced by topic segmentation. In: Proceedings of the 26th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR), pp. 306–313 (2003)
4. Bender, M., Michel, S., Triantafillou, P., Weikum, G.: Global document frequency estimation in peer-to-peer web search. In: Proceedings of 9th International Workshop on the Web and Databases (WebDB), pp. 62–67 (2006)
5. Cholvi, V., Felber, P., Biersack, E.: Efficient search in unstructured peer-to-peer networks. Tech. rep., Institut EURECOM (2003)
6. Clarke, I., Sandberg, O., Wiley, B., Hong, T.: Freenet: A distributed anonymous information storage and retrieval system. In: Proceedings of the ICSI Workshop on Design Issues in Anonymity and Unobservability, pp. 46–66 (2000)

7. Cohen, E., Fiat, A., Kaplan, H.: Associative search in peer-to-peer networks: Harnessing latent semantics. In: Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM) (2003)
8. Crespo, A., Garcia-Molina, H.: Routing indices for peer-to-peer systems. In: Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS), pp. 23– (2002)
9. Crespo, A., Garcia-Molina, H.: Semantic overlay networks for P2P systems. In: Proceedings of the 3rd International Workshop on Agents and Peer-to-Peer Computing (AP2PC), pp. 1–13 (2004)
10. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. Journal of the American Society of Information Science **41**(6), 391–407 (1990)
11. Doulkeridis, C., Nørvåg, K., Vazirgiannis, M.: The SOWES approach to P2P web search using semantic overlays. In: Proceedings of 15th International Conference on World Wide Web(WWW), pp. 1027–1028 (2006)
12. Doulkeridis, C., Nørvåg, K., Vazirgiannis, M.: DESENT: Decentralized and distributed semantic overlay generation in P2P networks. IEEE Journal on Selected Areas in Communications (J-SAC) **25**(1), 25–34 (2007)
13. Doulkeridis, C., Nørvåg, K., Vazirgiannis, M.: Peer-to-peer similarity search over widely distributed document collections. In: Proceedings of the 6th Workshop on Large-Scale Distributed Systems for Information Retrieval (LSDS-IR) (2008)
14. Doulkeridis, C., Vlachou, A., Kotidis, Y., Vazirgiannis, M.: Peer-to-peer similarity search in metric spaces. In: Proceedings of the 33rd International Conference on Very Large Data Bases(VLDB), pp. 986–997 (2007)
15. Eisenhardt, M., Mueller, W., Henrich, A.: Classifying documents by distributed P2P clustering. GI Jahrestagung pp. 286–291 (2003)
16. Ganti, V., Ramakrishnan, R., Gehrke, J., Powell, A.L., French, J.C.: Clustering large datasets in arbitrary metric spaces. In: Proceedings of the 15th International Conference on Data Engineering (ICDE), pp. 502–511 (1999)
17. Gnutella. http://en.wikipedia.org/wiki/Gnutella
18. Hui, K.Y.K., Lui, J.C.S., Yau, D.K.Y.: Small-world overlay P2P networks: Construction, management and handling of dynamic flash crowds. Computer Networks **50**(15), 2727–2746 (2006)
19. Klampanos, I.A., Jose, J.M.: An architecture for information retrieval over semi-collaborating peer-to-peer networks. In: Proceedings of the 2004 ACM Symposium on Applied Computing (SAC), pp. 1078–1083 (2004)
20. Koloniari, G., Pitoura, E.: Recall-based cluster reformulation by selfish peers. In: ICDE Workshops, pp. 200–205 (2008)
21. Kothari, A., Agrawal, D., Gupta, A., Suri, S.: Range addressable network: A P2P cache architecture for data ranges. In: Proceedings of the 3rd International Conference on Peer-to-Peer Computing (P2P), p. 14 (2003)
22. Li, J., Loo, B., J.M.Hellerstein, Kaashoek, M., Karger, D., Morris, R.: On the feasibility of peer-to-peer web indexing and search. In: Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS), pp. 207–215 (2003)
23. Li, M., Lee, W.C., Sivasubramaniam, A.: Semantic small world: An overlay network for peer-to-peer search. In: Proceedings of the 12th IEEE International Conference on Network Protocols (ICNP), pp. 228–238 (2004)
24. Linari, A., Weikum, G.: Efficient peer-to-peer semantic overlay networks based on statistical language models. In: Proceedings of Workshop on Information Retrieval in Peer-to-Peer Networks (P2PIR) (2006)
25. Liu, F., Li, M., Huang, L.: Distributed information retrieval based on hierarchical semantic overlay network. In: Proceedings of the 3rd International Conference on Grid and Cooperative Computing (GCC), pp. 657–664 (2004)

26. Lodi, S., Penzo, W., Mandreoli, F., Martoglia, R., Sassateli, S.: Semantic peer, here are the neighbors you want! In: Proceedings of 11th International Conference on Extending Database Technology (EDBT), pp. 26–37 (2008)
27. Löser, A.: Taxonomy-based overlay networks for P2P systems. In: Proceedings of the 8th International Database Engineering and Applications Symposium (IDEAS), pp. 407–412 (2004)
28. Löser, A., Naumann, F., Siberski, W., Nejdl, W., Thaden, U.: Semantic overlay clusters within super-peer networks. In: Proceedings of the 1st International Workshop on Databases, Information Systems, and Peer-to-Peer Computing (DBISP2P) (2003)
29. Löser, A., Staab, S., Tempich, C.: Semantic social overlay networks. IEEE Journal on Selected Areas in Communications **25**(1), 5–14 (2007)
30. Löser, A., Tempich, C.: On ranking peers in semantic overlay networks. In: Wissensmanagement, pp. 209–216 (2005)
31. Lv, J., Cheng, X.: WonGoo: A pure peer-to-peer full text information retrieval system based on semantic overlay networks. In: Proceedings of the 3rd IEEE International Symposium on Network Computing and Applications(NCA), pp. 47–54 (2004)
32. Michel, S., Triantafillou, P., Weikum, G.: Minerva∞: A scalable efficient peer-to-peer search engine. In: Proceedings of 6th International Middleware Conference, pp. 60–81 (2005)
33. Nejdl, W., Wolpers, M., Siberski, W., Schmitz, C., Schlosser, M., Brunkhorst, I., Löser, A.: Super-peer-based routing and clustering strategies for RDF-based P2P networks. In: Proceedings of the 12nd International Conference on World Wide Web (WWW), pp. 536–543 (2003)
34. Parreira, J.X., Michel, S., Weikum, G.: p2pDating: Real life inspired semantic overlay networks for web search. In: Proceedings of 2005 Workshop on Heterogeneous and Distributed Information Retrieval (HDIR) (2005)
35. Raftopoulou, P., Petrakis, E.G.M.: iCluster: A self-organizing overlay network for p2p information retrieval. In: Proceedings of the 30th European Conference on IR Research (ECIR), pp. 65–76 (2008)
36. Raftopoulou, P., Petrakis, E.G.M.: A measure for cluster cohesion in semantic overlay networks. In: Proceedings of the 6th Workshop on Large-Scale Distributed Systems for Information Retrieval (LSDS-IR) (2008)
37. Raftopoulou, P., Petrakis, E.G.M., Tryfonopoulos, C., Weikum, G.: Information retrieval and filtering over self-organising digital libraries. In: Proceedings of the 12th European Conference on Research and Advanced Technology for Digital Libraries (ECDL) (2008)
38. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Schenker, S.: A scalable content-addressable network. In: Proceedings of International Conference on Data Communication (SIGCOMM), pp. 161–172 (2001)
39. Ritter, J.: Why Gnutella can't scale. No, really! `http://www.darkridge.com/~jpr5/doc/gnutella.html` (2001)
40. Rowstron, A., Druschel, P.: Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In: Proceedings of the International Middleware Conference, pp. 329–350 (2001)
41. Schlosser, M., Sintek, M., Decker, S., Nejdl, W.: HyperCuP - Hypercubes, ontologies and efficient search on P2P networks. In: Proceedings of International Workshop on Agents and Peer-to-Peer Computing (AP2PC), pp. 112–124 (2002)
42. Sedmidubský, J., Barton, S., Dohnal, V., Zezula, P.: Adaptive approximate similarity searching through metric social networks. In: Proceedings of the 24th International Conference on Data Engineering (ICDE), pp. 1424–1426 (2008)
43. Shen, H.T., Shu, Y., Yu, B.: Efficient semantic-based content search in P2P network. IEEE Transactions on Knowledge and Data Engineering **16**(7), 813–826 (2004)
44. Sripanidkulchai, K., Maggs, B., Zhang, H.: Efficient content location using interest-based locality in peer-to-peer systems. In: Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM) (2003)
45. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. In: Proceedings of International Conference on Data Communication (SIGCOMM), pp. 149–160 (2001)

46. 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: Proceedings of 6th International Workshop on the Web and Databases (WebDB) (2003)
47. Tang, C., Xu, Z., Dwarkadas, S.: Peer-to-peer information retrieval using self-organizing semantic overlay networks. In: Proceedings of International Conference on Data Communication (SIGCOMM), pp. 175–186 (2003)
48. 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 (WWW), pp. 640–649 (2004)
49. Triantafillou, P., Xiruhaki, C., Koubarakis, M., Ntarmos, N.: Towards high performance peer-to-peer content and resource sharing systems. In: Proceedings of the 1st Biennial Conference on Innovative Data Systems Research (CIDR) (2003)
50. Vazirgiannis, M., Nørvåg, K., Doulkeridis, C.: Peer-to-peer clustering for semantic overlay network generation. In: Proceedings of the 6th International Workshop on Pattern Recognition in Information Systems (PRIS'06) (2006)
51. Vlachou, A., Doulkeridis, C., Mavroeidis, D., Vazirgiannis, M.: Designing a peer-to-peer architecture for distributed image retrieval. In: Proceedings of the 5th International Workshop on Adaptive Multimedial Retrieval: Retrieval, User, and Semantics (AMR), pp. 182–195 (2007)
52. Voulgaris, S., van Steen, M., Iwanicki, K.: Proactive gossip-based management of semantic overlay networks. Concurrency and Computation: Practice and Experience **19**(17), 2299–2311 (2007)
53. Yang, B., Garcia-Molina, H.: Designing a super-peer network. In: Proceedings of the 19th International Conference on Data Engineering (ICDE), pp. 49– (2003)
54. Zhao, B., Kubiatowicz, J., Joseph, A.: Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Tech. rep., U.C.Berkeley (2001)