Knowl Inf Syst (2007)
DOI 10.1007/s10115-006-0055-1

Knowledge and Information Systems

REGULAR PAPER

Peter Haase · Ronny Siebes · Frank van Harmelen

Expertise-based peer selection in Peer-to-Peer networks

Received: 11 August 2005 / Revised: 10 April 2006 / Accepted: 3 November 2006

Abstract Peer-to-Peer systems have proven to be an effective way of sharing data. Modern protocols are able to efficiently route a message to a given peer. However, determining the destination peer in the first place is not always trivial. We propose a model in which peers advertise their expertise in the Peer-to-Peer network. The knowledge about the expertise of other peers forms a semantic topology. Based on the semantic similarity between the subject of a query and the expertise of other peers, a peer can select appropriate peers to forward queries to, instead of broadcasting the query or sending it to a random set of peers. To calculate our semantic similarity measure, we make the simplifying assumption that the peers share the same ontology. We evaluate the model in a bibliographic scenario, where peers share bibliographic descriptions of publications among each other. In simulation experiments complemented with a real-world field experiment, we show how expertise-based peer selection improves the performance of a Peer-to-Peer system with respect to precision, recall and the number of messages.

Keywords P2P · Routing · Semantic overlays · Ontologies

1 Introduction

Peer-to-Peer systems are distributed systems without centralized control or hierarchical organization, in which each node runs software with equivalent functionality. A review of the features of recent Peer-to-Peer applications yields a long list: redundant storage, permanence, selection of nearby servers, anonymity, search,

P. Haase
Institute AIFB, University of Karlsruhe, 76128 Karlsruhe, Germany

R. Siebes (✉) · F. van Harmelen
Department of Computer Science, Vrije Universiteit Amsterdam, De Boelelaan 1081, 1081HV, The Netherlands
E-mail: ronny@siebes.net

authentication, and hierarchical naming. Despite this rich set of features, scalability is a significant challenge: Peer-to-Peer networks that broadcast all queries to all peers do not scale—intelligent query routing and network topologies are required to be able to route queries to a relevant subset of peers. Modern routing protocols like Chord [25], CAN [22] and Pastry [23] are based on Distributed Hash Tables for efficient query routing, but little effort has been made with respect to rich semantic representations of metadata and query functionalities beyond simple keyword searches.

The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation [4]. In a Peer-to-Peer system, Semantic Web techniques can be used for expressing the knowledge shared by peers in a well-defined and formal way. In the simple model that we propose, peers use a shared ontology to advertise their expertise in the Peer-to-Peer network. The knowledge about the expertise of other peers forms a semantic overlay network, independent of the underlying network topology. If a peer receives a query, it can decide to forward it to peers about which it knows that their expertise is *similar* to the subject of the query. The advantage of this approach is that queries will not be forwarded to all or a random set of known peers, but only to those that have a good chance of answering it.

In this paper, we instantiate the above model with a bibliographic scenario, in which researchers share bibliographic metadata about publications. We present results of both simulation experiments and a real-world field experiment.

In the evaluation using the simulation experiments of our model we show how

- the proposed model of expertise-based peer selection considerably improves the performance of the Peer-to-Peer system,
- ontology-based matching with a similarity measure improves the system compared with an approach that relies on exact matches, such as a simple keyword based approach,
- the performance of the system can be improved further, if the semantic overlay network is built according to the semantic similarity of the expertise of the peers,
- a "perfect" semantic overlay network imposed on the network using global knowledge yields ideal results.

The results from the field experiment with the Bibster system validate the applicability and performance of the model for real-world systems.

In the remainder of the paper, we discuss related work (Sect. 2), present the formal model for expertise-based peer selection (Sect. 3), instantiate this model for the bibliographic scenario (Sect. 4), define evaluation criteria (Sect. 5), present results of the simulation experiments (Sect. 6) and the field experiment (Sect. 7), and conclude with some directions for future work (Sect. 8).

2 Background and related work

Peer-to-Peer systems are typically characterized by the absence of a single central instance of control. This has consequences for the network organization and the coordination to route requests to the experts able to respond to the request. Peer

selection plays a role in all Peer-to-Peer systems that are dealing with document discovery. By definition, any such system must have *a* strategy for peer selection (even if it is only a trivial network strategy), and many systems try to improve on this in order to avoid network congestion.

In completely unstructured Peer-to-Peer networks, the data is distributed randomly, and broadcasting mechanisms are used to distribute queries. In structured networks networks, a distributed index is built to route search requests. This structure can involve various degrees of central coordination or global knowledge, e.g. relying on super-peers. Further, we can distinguish whether the indexing structure relies on exact (syntactic) matches of keys to route requests, or whether they consider the semantics of the request.

Although many real systems which are concerned on finding expertise make use of approaches that combine developments from different research fields, they will more or less fit or be a combination of one of the following techniques:

2.1 Broadcasting

Although a very simple technique, broadcasting has already proven its usefulness in small networks and in larger Peer-to-Peer file-sharing systems [14]. The idea is that peers keep forwarding a query to their neighbors until a sufficient number of answers is found or till maximum number of forwards (hops) are reached. This approach is not very scalable, because a query can result in a large number of messages which consumes an unacceptable usage of network capacity. Also, it is possible that even if the data is somewhere in the network it will not be found due to the maximum number of hops. The big advantage of broadcasting approaches is that they have very low maintenance costs and dependency, meaning that almost no messages are needed to keep the network alive and that the network is very robust to frequent peer drops and joins (network dynamics). In case where broadcasting really is needed, Hypercup [24] guarantees that only $O(N-1)$ messages and $O(\log(N))$ hops are needed to reach all peers, where N is the number of peers in the network. Moreover, they show how their scheme can be made even more efficient by using a global semantic network to determine the organization of peers in the graph topology. Namely, when peers describe their content in terms of this shared data-structure, peers are able to cluster themselves with similar peers. This approach based on a structured hypercube overlay has more maintenance overhead and is therefore also more sensitive to network dynamics than traditional broadcasting approaches.

2.2 Central registries

An easy but not very robust approach is to have a single register where systems can advertise their expertise descriptions or to have the registry itself search the network for expertise descriptions. A well-known example from the Peer-to-Peer community, but only partially Peer-to-Peer, is Napster.[1] This system has one large

[1] Napster. http://www.napster.com/about_us.html, 2002.

repository which combines filenames with peers that offer those files for downloading. Such a repository can be seen as yellow pages, where each member in the network can look up the person or system that fulfills its needs. In small organizations, such an approach could work very well because the network is small and stable, so that the registry does not have to do much query processing and updates. In larger networks the approach is not very robust and has the same disadvantages as completely centralized approaches: undisclosed content, scalability problems, lack of privacy and censor possibilities.

2.3 Brokering

The Multi-Agent community suggested the concept of 'broker agents' like in InfoSleuth [13], which semantically match information needs (specified in terms of some shared data-structure, e.g. an ontology) with currently available resources which are found by the broker itself or registered by the providing agents. In InfoSleuth, agents advertise their services to the broker via the KQML [10] language. Broker agents respond to an agent's request for service with information about the other agents that have previously advertised relevant services. The literature on broker agents has a clear focus on finding services. Therefore, it is not surprising that the brokering approach is very popular in the literature on finding web-services which are semantically described [19]. One thing where the literature is not clear about is on how scalable and robust this approach is. In a network where millions of agents offer their services, one broker agent probably will not be enough and will have the same problems as with a central registry.

2.4 Super-peers/nodes

An approach that looks very similar to brokering but with a different goal in mind, comes from the Peer-to-Peer research community. The technique, which works well for file sharing, makes use of the different capacities of the nodes in a Peer-to-Peer network: Peers that have more processing power, memory or network bandwidth than other peers are assigned additional tasks in the network. For example, KaZaa [16] lets peers voluntary act as super-peers that maintain large routing tables, in which information is stored about the content of other peers (comparable to yellow pages). Relying on super-peers, this approach introduces a form of centralization in the system. Although better than broadcasting in a network without super-nodes, this remains essentially broadcasting and therefore can be improved by techniques that do more efficient routing described in the next paragraphs.

Nejdl et al. [20] presents schema-based Peer-to-Peer networks and the use of super-peer-based topologies for these networks, in which peers are organized in hypercubes. This topology guarantees that each node is queried exactly once for each query. Löser et al. [18] shows how this schema-based approach can be used to create Semantic Overlay Clusters in a scientific Peer-to-Peer network with a small set of metadata attributes that describe the documents in the network. In

contrast, the approach in our system is completely decentralized in the sense that it does not rely on super-peers.

2.5 Distributed hash tables and distributed search trees

Another technique that comes from the Peer-to-Peer research community makes use of Distributed Hash Tables (DHT). DHTs are based on the idea to route content (or a pointer to the content) to the peer whose identifier lies closest to the unique identifier of the content. This technique assumes that all peers have the same 'hash' function to assign a unique (mostly 128 bit) identifier to content, which could be anything like documents, music, URLs or words. The characteristic of this technique is that it allows to route content and queries in $O(\log(n))$ steps to the right peers, where n is the number of peers in the network. Also, systems that do routing based on DHTs, such as Chord [25] and Pastry [23], are robust with respect to rapid join and leaves of peers. A disadvantage of most DHT approaches is that they have high maintenance costs, due to the frequent changes in the overlay network as a result of peers continuously joining and leaving. P-Grid [1] is a Peer-to-Peer search system based on a virtual distributed search tree, similarly structured as standard distributed hash tables, but with an unstructured way of building the DHT-overlay. Namely, P-Grid uses randomized algorithms for constructing the access structure, updating the data and performing search. In this way, probabilistic estimates can be given for the success of search requests, and search is more robust than the previously described DHT approaches against failures of nodes. A disadvantage of all DHT approaches is that objects that are not hashed cannot be found, which is a problem for full-text searching. To be specific, in a document sharing case, one could roughly do two things: (1) The file itself is hashed to a unique key. The disadvantage is that the user has to know this key too, which is highly unrealistic. (2) The title of the document is hashed. This is still a problem because one type error would result in a complete different hash key. (3) All the words in the document are hashed and the document or the location of the document is stored at the peers on which the identifiers are closest to the hash keys of the words. Although now someone is able to find the documents that contain the keywords, the procedure of distributing the hash keys is not efficient because all these keys have to be distributed to the right peers in the network. Another disadvantage of a pure DHT-based approach is that load-balancing is not an emergent property of the topology. Due to the fact that content and queries follow a power law distribution, some peers (responsible for popular keys) are much more loaded than other peers that accidently are responsible for less popular ones. Therefore, active load-balancing strategies have to be developed on top of DHT, which is not needed for broadcast-based and expertise-based (described in next paragraph) alike approaches. Also, a pure DHT-based approach is less robust than broadcast-based and expertise-based approaches, because normally only one peer is responsible for one key, and if that peer does not respond to queries (for example, behind a fire-wall or due to overload), no content can be found that is hashed to that key. The work of Byers et al. [6] confirms the load-balancing and bottleneck problem and describes an alternative DHT approach to solve it by introducing redundancy of content pointers in the network, which however generates significant additional maintenance costs.

2.6 Semantic overlay networks

Peers that keep pointers to other peers which have similar content to themselves form a Semantic Overlay Network (SON). Gridvine [2] provides semantic overlay network on top of PGrid: While PGrid as a structured Peer-to-Peer network for efficient routing of messages provides the 'physical' layer, Gridvine introduces a semantic overlay for managing and mapping data and metadata schemas as the 'logical' layer. In essence, the efficiency of the search algorithm is caused not by smart forwarding queries based on the semantic overlay, but by applying the underlying DHT approach for mapping terms to peers.

Because of the focus of our own work on semantic topologies, we look closer at systems where the goal is an efficient search mechanism based on routing queries to peers that are semantically closest to the content of the query.

One approach to achieve that is to classify the content of a peer into a shared topic vector where each element in the vector contains the relevance for that given peer for the respective topic. pSearch [26], is such an example where documents in the network are organized around their vector representations (based on modern document ranking algorithms) such that the search space for a given query is organized around related documents, achieving both efficiency and accuracy. In pSearch, for each element in the topic vector, each peer has a responsibility for a certain range or interval, e.g. ($[0.2 - 0.4], [0.1 - 0.3]$). Now all expertise vectors that fall in that range are routed to that peer, meaning that, following the example vector, the expertise vector $[0.23, 0.19]$ would be routed to this peer and $[0.13, 0.19]$ not. Besides, the responsibility for a vector range, a peer also knows the list of neighbors which are responsible to vector ranges close to itself. The characteristic of pSearch is that the way that peers know about close neighbors is very efficient. A disadvantage of pSearch is that all documents have to be mapped into the same (low dimensional) semantic search space and that the dimensionality on the overlay is strongly dependent of the dimensionality of the vector, with the result that each peer has to know many neighbors when the vectors have high a dimension.

Another approach is based on random walk clustering [28], where peers with similar content are going to know each other. The assumption is that queries posted by (the users of) peers are semantically closely related to the content of the peer itself. This results in a high probability that the neighbors of the peer (the peers in the cluster of that peer) have answers to the query. The problem of this approach in the domain of full-text searches, is what information a peer has to tell to another peer so that they are able to determine if they are related or not. When there is no shared data-structure (like a fixed set of terms) in which they can describe their content, the whole content has to be shared. This results in the fact that much data has to be shared between peers for determining closeness.

Caching of pointers to popular content based on query answers is done in Freenet [7]. In short, when a node forwards a request for a particular key to another node in the network, and that node is successful in retrieving the data, the address of an upstream node (possibly the one where the data originated) is included in the reply. The requester makes a note of the requested key, and the source node passed back with that reply. It is assumed that the upstream node is a good place to route future requests for keys closest to the previously requested key.

There is also work on ‘routing indices’ where a peer maintains knowledge about the reachable content from its neighbors. For example, the work of [8] describes a method where peers summarize their knowledge in a set of topics and advertise this with the number of documents that they can reach to their direct neighbors. With ‘reaching’ the authors mean that the peer itself has documents on that topic, or knows other peers that have such documents. The problem with this approach is that either these index tables are very large (resulting in expensive maintenance because these indexes are sent to neighbors when updates occur) or are not rich enough to have an overlap of tables between peers, resulting in dead-ends a forwarding process.

In contrast to the previous approach, the last SON approach that we discuss here lets peers describe their content in a shared set of terms. Mostly, these terms are organized in a topic network or hierarchy making it able to determine the semantic similarity between terms. Each peer is characterized by a set of topics that describe its expertise. A peer knows about other peer’s expertise topics by analyzing advertisement messages [12] or answers [27]. In this way, peers form clusters of semantic related expertise descriptions. Given a query, a shared distance metric allows to forward queries (described by a shared set of terms) to neighbors whose expertise description is semantically closely related to the query. The advantages of this approach are threefold:

– *Peer autonomy* Each peer can, in principle, have its own distance measure, peer selection mechanism and advertisement strategy. This allows peers, for example to keep their neighbor list or similarity metric secret. Also peers can decide at any time to change their visibility on the network by sending advertisement messages.
– *Automatic load-balancing* When some content is provided by many peers also the semantic cluster on that content will contain many peers. In this way, load-balancing is an emergent property of this approach.
– *Robustness/fault tolerance* When peers leave the network or do not respond to a query, the only consequence is that they probably will not be asked a next time until they send new advertisement messages or are recommended by other peers. In contrast, most DHT approaches have to move routing tables to other peers in order to restore the overlay.

However, there is also a disadvantage, terms that are not shared can not be found. For example, imagine that a peer has some documents containing the phrase ‘database languages’, but the shared data-structure only contains the term ‘databases’, then two things can be done (1) extend the shared data-structure with the word ‘database languages’ so that peers are able to query and describe their expertise with that term or (2) the functions that extracts the expertise description and abstract the queries should be intelligent enough to see that ‘databases’ is a good replacement for ‘database languages’. Note that in this case the original query still contains ‘database languages’, but the routing mechanism uses the shared term ‘databases’ to route it to the peer that registered itself on that term. Both solutions have their own problems, the first one will lead eventually to very large data-structures, the second one depends very heavily on the quality of the extraction and abstraction algorithms.

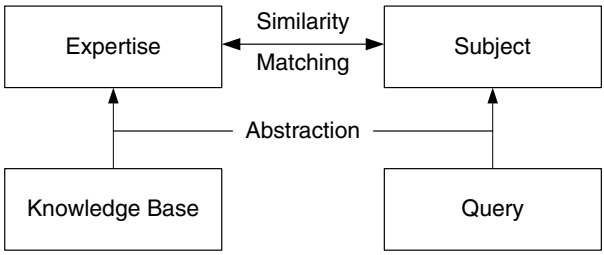


Fig. 1 Expertise-based matching

3 A model for expertise-based peer selection

In the model that we propose, peers advertise their expertise in the network. The peer selection is based on matching the subject of a query and the expertise according to their semantic similarity. Figure 1 below shows the idea of the model in one picture. Our model is deliberately simple, in order to make as few assumptions as possible about the architecture of both the network and the individual peers, so as to make our work as widely applicable as possible.

In this section, we first introduce a model to semantically describe the expertise of peers and how peers promote their expertise as advertisement messages in the network. Second, we describe how the received advertisements allow a peer to select other peers for a given query based on a semantic matching of query subjects against expertise descriptions. The third part describes how a *semantic overlay network* can be formed by advertising expertise.

3.1 Semantic description of expertise

3.1.1 Peers

The Peer-to-Peer network consists of a set of peers P. Every peer $p \in P$ has a knowledge base that contains the knowledge that it wants to share.

3.1.2 Common ontology

The peers share an ontology O, which provides a common conceptualization of their domain by defining a set of terms and the relations between them. The ontology is used for describing the expertise of peers and the subject of queries. Although we assume that all peers share the same ontology, it can be expected that a partial overlap between different ontologies would give similar results. Distributing the ontology to all peers can be done when the user downloads the application.

3.1.3 Expertise

An expertise description $e \in E$ is a abstract, semantic description of the knowledge base of a peer based on a set of terms from the common ontology O. This expertise can either be extracted from the knowledge base automatically or specified in some other manner.

3.1.4 Advertisements

Advertisements $A \subseteq P \times E$ are used to promote descriptions of the expertise of peers in the network. An advertisement $a \in A$ associates a peer p with an expertise description e.

3.1.5 Advertisement distribution algorithm

Peers decide autonomously, without central control, whom to promote advertisements to and which advertisements to accept. This decision can be based on the semantic similarity between expertise descriptions.

3.2 Matching and peer selection

We now turn to the discussion how peers are selected based on a given query using a similarity function to rank peers.

3.2.1 Queries

Queries $q \in Q$ are posed by a user and are evaluated against the knowledge bases of the peers. First, a peer evaluates the query against its local knowledge base and then decides which peers the query should be forwarded to. Query results are returned to the peer that originally initiated the query.

3.2.2 Subjects

A subject $s \in S$ is an abstraction of a given query q expressed in a set of terms from the common ontology O. The subject can be seen as a complement to an expertise description, as it specifies the required expertise to answer the query.

3.2.3 Similarity function

The similarity function $SF_S : S \times E \mapsto [0, 1]$ yields the semantic similarity between a subject $s \in S$ and an expertise description $e \in E$. An increasing value indicates increasing similarity. If the value is 0, s and e are not similar at all, if the value is 1, they match exactly. SF_S is used for determining to which peers a query should be forwarded. Analogously, a same kind of similarity function $SF_E : E \times E \mapsto [0, 1]$ can be defined to determine the similarity between the expertise of two peers.

3.2.4 Peer selection algorithm

The peer selection algorithm returns a ranked set of peers. The rank value is equal to the similarity value provided by the similarity function.

From this set of ranked peers one can, for example, select the best n peers, or all peers whose rank value is above a certain threshold, etc.

3.3 Semantic overlay network

The knowledge of the peers about the expertise of other peers is the basis for a semantic topology. It is important to state that this semantic topology is independent of the underlying network topology. At this point, we make no assumptions about the topology of the network.

The semantic overlay network can be described by the following relation:

Knows $\subseteq P \times P$, where *Knows*(p_1, p_2) means that p_1 knows about the expertise of p_2.

The relation *Knows* is established by the selection of which peers a peer sends its advertisements to and from which peers a peer accepts advertisements. The semantic overlay network in combination with the expertise-based peer selection is the basis for intelligent query routing. The intuition of the overlay network is to establish acquaintances between peers with similar expertise in order to be able to route queries along a short path of increasing similarity between the subject of the query and the expertise of the peers. Different strategies for establishing such acquaintances will be presented and evaluated in the following sections.

3.4 Consequences of the model

An important value of the model described above is that it dictates which design decisions must be made when equipping a Peer-to-Peer network with expertise-based peer selection. These decisions are as follows:

- We must define the *ontology* as a set of terms and a set of relations between them.
- We must define *two abstraction functions*: one to abstract the contents of peers to expertise descriptions (sets of terms from the ontology), and one to abstract queries to subjects (again sets of terms from the ontology).
- We must define *two advertisement policies*: to which peers should advertisements be sent, and which advertisements should be accepted.
- We must define *two similarity functions*: one to compare subjects with expertise descriptions, and one to compare expertise descriptions with each other.
- We must define a *peer selection algorithm* to decide to which peers queries must be routed.

We believe this model to be of general value in understanding Peer-to-Peer models with semantic query routing.

4 The bibliographic scenario

In this section, we instantiate the general model for expertise-based peer selection from previous section. We use a real-life scenario for knowledge sharing in a Peer-to-Peer environment.

In the daily life of a computer scientist, one regularly has to search for publications or their correct bibliographic metadata. Currently, people do these searches

with search engines like Google and CiteSeer, via university libraries or by simply asking other people that are likely to know how to obtain the desired information.

The scenario that we envision here is that researchers in a community share bibliographic metadata via a Peer-to-Peer system. The data may have been obtained from BibTeX files or from a bibliography server such as the DBLP database.[2] A similar scenario is described in [3], where data providers, i.e. research institutes, form a Peer-to-Peer network which supports distributed search over all the connected metadata repositories.

We now describe the bibliographic scenario using the general model presented in the previous section.

5 Peers

A researcher is represented by a peer $p \in P$. Each peer has an RDF [15] knowledge base, which consists of a set of bibliographic metadata items that are classified according to the ACM topic hierarchy.[3] The following example shows a fragment of a sample bibliographic item based on the Semantic Web Research Community Ontology (SWRC)[4]:

```
<rdf:RDF xmlns=
 "http://www.semanticweb.org/ontologies/swrc-onto.daml#"
  xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:acm ="http://daml.umbc.edu/ontologies/topic-ont#">
<Publication rdf:about="dblp:persons/Codd81">
 <title>The Capabilities of
        Relational Database Management Systems.</title>
 <acm:topic rdf:resource=
   "http://daml.umbc.edu/ontologies/classification#
    ACMTopic/Information_Systems/Database_Management"/>
 <!-- ... -->
</Publication>
</rdf:RDF>
```

5.1 Common ontology

The ontology O that is shared by all the peers is the ACM topic hierarchy. The topic hierarchy contains a set, T, of 1287 topics in the computer science domain and relations ($T \times T$) between them: *SubTopic* and *seeAlso*. It is important to state that this topic hierarchy is not an 'ISA' hierarchy, but a generalization/specialization organized tree structure. If it were an ISA hierarchy, experts on a topic would also be experts on all sub-topics. This is not the case in our situation, because experts could have expertise on a very specific topic, but do not have much generic knowledge on a super-topic standing high in the hierarchy. For example, imagine an expert on *Robot Sensoring by using Bayesian Classifiers*, which is a sub-topic of *Artificial Intelligence*. This expert does not need to have any expertise on AI in general at all. This means that our topic hierarchy cannot be used for inferring expertise by inheritance over the sub-topic relation. Instead, we use a similarity measure to calculate the semantic distance between topics.

[2] http://dblp.uni-trier.de/.

[3] http://daml.umbc.edu/ontologies/classification.

[4] http://ontoware.org/projects/swrc/.

5.2 Expertise

The ACM topic hierarchy is the basis for our expertise model. Expertise E is defined as $E \subseteq 2^T$, where each $e \in E$ denotes a set of ACM topics, for which a peer provides classified instances.

5.3 Advertisements

Advertisements associate peers with their expertise: $A \subseteq P \times E$. A single advertisement therefore consists of a set of ACM topics to which the peer is an expert.

5.4 Advertisement distribution algorithm

To keep the set of simulation parameters within acceptable boundaries, we choose the simple solution of letting a peer to send its advertisement only to its direct neighbors. We therefore do not use any advertisement forwarding policy. We do however simulate different advertisement acceptance policies, which are described in one of the paragraphs from the next section on the simulation settings. The average maintenance costs of a semantic overlay can be derived by multiplying the average frequency of advertising times the average number of peers in the network.

5.5 Queries

We use the RDF query language SeRQL [5] to express queries against the RDF knowledge base of a peer. The following sample query asks for the titles of publications whose ACM topic is *Information Systems* / *Database Management*:

```
CONSTRUCT {pub} <swrc:title> {title} FROM
{Subject} <rdf:type> {<swrc:Publication>};
  <swrc:title> {title};
  <acm:topic>
  {<topic:ACMTopic/Information_Systems/Database_Management>}
USING NAMESPACE
swrc=<!http://www.semanticweb.org/ontologies/swrc-onto.daml#>,
rdf=<!http://www.w3.org/1999/02/22-rdf-syntax-ns#>,
acm=<!http://daml.umbc.edu/ontologies/topic-ont#>,
topic=<!http://daml.umbc.edu/ontologies/classification#>
```

5.6 Subjects

Analogously to the expertise, a subject $s \in S$ is an abstraction of a query q. In our scenario, $S \subseteq 2^T$ each s is a set of ACM topics, thus $s \subseteq T$. For example, the extracted subject of the query above would be {*Information Systems*/*Database Management*}.

5.7 Similarity function

In this scenario, we use one similarity function SF ($SF = SF_E = SF_S$), which is based on the idea that topics which are close according to their positions in the topic hierarchy are more similar than topics that have a larger distance. For example, an expert on ACM topic *Information Systems*/*Information Storage and Retrieval* has a higher chance of giving a correct answer on a query about *Information Systems*/*Database Management* than an expert on a less similar topic like *Hardware*/*Memory Structures*. To be able to define the similarity of a peer's expertise and a query subject, which are both represented as a set of topics, we first define the similarity for individual topics. Reference [17] have compared different similarity measures between words in WordNet, based on the hyponym relations between them. Given that the hyponym structure is a hierarchically structured generality/specificity network, we assume that this metric also applicable to our ACM topic hierarchy. Their best performing similarity measure that gave the best results on their data set is as follows:

$$S(t_1, t_2) = \begin{cases} e^{-\alpha l} \cdot \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}} & \text{if} \quad t_1 \neq t_2, \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

Here l is the length of the shortest path between topic t_1 and t_2 in the graph spanned by the *SubTopic* relation. h is the level in the tree of the lowest common subsumer from t_1 and t_2; $\alpha \geq 0$ and $\beta \geq 0$ are parameters scaling the contribution of shortest path length l and depth h, respectively. Based on benchmark data from [17], the optimal values are: $\alpha = 0.2$, $\beta = 0.6$. Using the shortest path between two topics is a measure for similarity because Rada et al. [21] have proven that the minimum number of edges separating topics t_1 and t_2 is a metric for measuring the conceptual distance of t_1 and t_2. The intuition behind using the depth of the direct common subsumer in the calculation is that topics at upper layers of hierarchical semantic nets are more general and are semantically less similar than topics at lower levels. Our sub-topic hierarchy is a tree structure, but the metric from [17] is also able to deal with DAG (Directed Acyclic Graph) structures in general, by selecting the shortest path between two topics of interest.

Now that we have a function for calculating the similarity between two individual topics, we define SF as:

$$SF(s, e) = \frac{1}{|s|} \sum_{t_i \in s} \max_{t_j \in e} S(t_i, t_j) \tag{2}$$

This function iterates over all topics t_i of the subject s and averages their similarities with the most similar topic of the expertise e.

5.8 Peer selection algorithm

The peer selection algorithm ranks the known peers according to the similarity function described above. Therefore, peers that have an expertise more similar to that of the subject of the query will have a higher rank. From the set of ranked peers, we now only consider a selection algorithm that selects the best n peers.

To prevent cycles in the forwarding loop, each query message is identified by a unique identifier and each peer only responds to each unique query only once. The costs of the algorithm in terms of the number of forwarded query messages is an experimental variable, for which the results are shown in the next sections on the simulation and field experiment.

We have now made a decision on many of the points dictated by the general model from the previous section: a common ontology, expertise and query-subject descriptions, advertisement-contents, and similarity functions. Still missing are the advertisement policy, used for propagating expertise, and the abstraction functions, used for describing content and queries. These are experimental variables because we test different policies, and therefor will be discussed in Sect. 6, where we describe the details of our experiments.

6 Evaluation criteria

In this section, we define a number of criteria for a Peer-to-Peer system, which will be the basis for the evaluation of our proposed model for peer selection. These criteria are mainly based on those described in [9]. We distinguish between input parameters *affect* the performance of the system, and output parameters that *are affected* and serve as measures for the performance of the system.

6.1 Input parameters

The following input parameters are important criteria that influence the performance of a Peer-to-Peer system:

6.1.1 Number of peers

The size of the Peer-to-Peer network is represented by this number. Typically, the scalability of the system is measured in terms of number of peers. The number of peers varies depending on the distribution of documents.

6.1.2 Number of documents

The scalability of a Peer-to-Peer system can also be expressed in terms of the number of shared resource items, e.g. documents.

6.1.3 Document distribution

The document distribution in Peer-to-Peer networks is rarely completely random, but often has certain properties. With this input parameter, we want to evaluate how the proposed model behaves with different document distributions.

6.1.4 Network topology

The performance of a Peer-to-Peer system is strongly influenced by the network topology and its characteristics. Possible topologies could for example be super-peer based, star or ring-shaped, or simply a random graph.

6.1.5 Advertisement policy

The advertisements are responsible for building the semantic overlay network. There are various variables involved, e.g. whom to send the advertisements to and which received advertisements to accept based on the semantic similarity between the own expertise and that of the advertisement.

6.1.6 Peer selection algorithm

The peer selection algorithm determines which peers a query should be forwarded to. This could be a naive algorithm, which simply broadcasts a query, or a more advanced one, as the proposed expertise-based peer selection.

6.1.7 Maximum number of hops

The maximum number of hops specifies how many times a query is allowed to be forwarded. It determines how much the network will be flooded by a single query.

6.2 Output parameters

To evaluate a Peer-to-Peer system, we use precision and recall measures known from classical Information Retrieval. Here, we distinguish measures on the document level (query answering) and the peer level (peer selection). Note that for our simulation of the bibliographic scenario we disregard the actual documents (i.e. papers) and only distribute their metadata (i.e. their bibliographic descriptions). These measures are defined as follows:

Document level (Query Answering).

$$Precision_{Doc} = \frac{|Docs_{relevant} \bigcap Docs_{returned}|}{|Docs_{returned}|}$$

indicates how many of the returned documents are relevant, with $Doc_{relevant}$ being the set of relevant documents in the network, meaning that the terms in the query match their metadata description, and $Docs_{returned}$ being the set of returned documents. We determine the set of relevant documents $Docs_{relevant}$ by evaluating the query against a centralized database which contains the complete data set. In our model, we work with exact queries, therefore only relevant documents are returned. The precision will hence always be one,meaning that the document pre-

cision is not a useful measure to use:

$$Precision_{Doc} = \frac{|Docs_{returned}|}{|Docs_{returned}|} = 1.$$

$$Recall_{Inf} = \frac{|Docs_{relevant} \bigcap Docs_{returned}|}{|Docs_{relevant}|} = \frac{|Docs_{returned}|}{|Docs_{relevant}|}$$

The recall on the document level states how many of the relevant documents are returned.

Peer Level (Peer Selection).

$$Precision_{Peer} = \frac{|Peers_{relevant} \bigcap Peers_{reached}|}{|Peers_{reached}|}$$

For a given query, how many of the peers that were selected had relevant information. Here, $Peers_{relevant}$ is the set of peers that had relevant documents and $Peers_{reached}$ is the set of peers that were reached.

$$Recall_{Peer} = \frac{|Peers_{relevant} \bigcap Peers_{reached}|}{|Peers_{relevant}|} = \frac{|Peers_{reached}|}{|Peers_{relevant}|}$$

indicates for a given query, how many of the peers that had relevant information were reached.

6.2.1 Further parameters

Another important output parameters is:

$$Number_{Messages}$$

This output parameter indicates with how many messages the network is flooded by one query. The number of messages does not only affect the network traffic, but also CPU consumption, such as for the processing of the queries in the case of query messages.

There are many other output parameters that we could have used as additional evaluation criteria. Examples are the size of messages between peers, the response times on queries to the network, CPU load of individual peers, etc. However, we do not report on these as they are not relevant to our evaluation hypotheses and therefore also not captured by our simulation software.

7 Simulation experiments

In this section, we describe the simulation of the scenario presented in Sect. 4. The evaluations are based on the criteria defined in Sect. 6. With the experiments we validate the following hypotheses:

Hypothisis 1 (Expertise-based selection) *The proposed approach of expertise-based peer selection yields better results than a naive approach based on random selection. The higher precision of the expertise-based selection results in a higher recall of peers and documents, while reducing the number of messages per query.*

Hypothisis 2 (Ontology-based matching) *Using a shared ontology with a metric for semantic similarity improves the recall rate of the system compared with an approach that relies on exact matches, such as a simple keyword based approach.*

Hypothisis 3 (Semantic overlay network) *The performance of the system can be improved further, if the semantic topology is built according to the semantic similarity of the expertise of the peers. This can be realized, for example, by accepting advertisements that are semantically similar to the own expertise.*

Hypothisis 4 (The "Perfect" overlay network) *Perfect results in terms of precision and recall can be achieved, if the semantic overlay network coincides with a distribution of the documents according to the expertise model.*

7.1 Setup of the simulation experiments

In the following, we describe the setup of the simulation experiments performed: the data sets used, the distribution of the data, the simulation environment, and the individual experimental settings.

7.2 Data set

To obtain a critical mass of bibliographic data, we used the DBLP data set, which consists of metadata for 380,440 publications in the computer science domain.

We have classified the publications of the DBLP data set according to the ACM topic hierarchy using a simple classification scheme based on lexical analysis: A publication is said to be about a topic, if the label of the topic occurs in the title of the publication. For example, a publication with the title "The Capabilities of Relational Database Management Systems." is classified into the topic *Database Management*. Topics with labels that are not unique (e.g. *General* is a sub-topic of both *General Literature* and *Hardware*) have been excluded from the classification, because typically these labels are too general and would result in publications classified into multiple, distant topics in the hierarchy. Obviously, this method of classification is not as precise as a sophisticated or manual classification. However, a high precision of the classification is not required for the purpose of our simulations. As a result of the classification, about one third of the DBLP publications (126,247 out of 380,440) have been classified, against 553 out of the 1287 ACM topics. The classified DBLP subset has been used for our simulations.

7.3 Document distribution

We have simulated and evaluated the scenario with two different distributions, which we describe in the following. Note that for the simulation of the scenario we disregard the actual documents and only distribute the bibliographic metadata of the publications.

7.4 Topic distribution

In the first distribution, the bibliographic metadata are distributed according to their topic classification. There is one dedicated peer for each of the 1287 ACM topics. The distribution is directly correlated with the expertise model, each peer is an expert on exactly one ACM topic and contains all the corresponding publications. This also implies that there are peers that do not contain publications, because not all topics have classified instances.

7.5 Proceedings distribution

In the second distribution, the bibliographic metadata are distributed according to conference proceedings and journals in which the according publications were published. For each of the conference proceedings and journals covered in DBLP, there is a dedicated peer that contains all the associated publication descriptions (in the case of the 328 journals) or inproceedings (in the case of the 2006 conference proceedings). Publications that are published neither in a journal nor in conference proceedings are contained by one separate peer. The total number of peers therefore is $2335 (= 328 + 2006 + 1)$. With this distribution, one peer can be an expert on multiple topics, as a journal or conference typically covers multiple ACM topics. Note that there is still a correlation between the distribution and the expertise, as a conference or journal typically covers a coherent set of topics.

We do not make any assumptions on how these distributions are achieved, so we see them as given in our simulations. One way to distribute content in this way is via DHT where the keys are topics or conference identifiers, so that each of them is mapped to a unique peer in the network. We already mentioned some problems with DHT approaches such as no load-balancing and single points of failures. Our experiments can be seen as a way to investigate how semantic methods can be used to mitigate some of these problems.

7.6 Simulation environment

To simulate the scenario we have developed and used a controlled, configurable Peer-to-Peer simulation environment. A single simulation experiment consists of the following sequence of operations:

1. *Setup network topology:* In the first step, we create the peers with their knowledge bases according to the document distribution and arrange them in a random network topology, where every peer knows 10 random peers. We have fixed this number in our simulations to keep the number of different variable tractable, and have chosen this value to simulate a realistic sparse topology. We do not make any further assumptions about the network topology.
2. *Advertising knowledge:* In the second step, the semantic overlay network is created. Every peer sends an advertisement of its expertise to all other peers it knows based on the overlay network. When a peer receives an advertisement, it may decide to store all or only selected advertisements, e.g. if the advertised expertise is semantically similar to its own expertise. After this step, the semantic overlay network is static and will not change anymore.

3. *Query Processing:* The peers randomly initiate queries from a set of randomly created 12,870 queries, 10 for each of the 1287 ACM topics. The peers first evaluate the queries against their local knowledge base and then propagate the query according to their peer selection algorithms described below.

We currently do not simulate any node drops and node joins, which would be needed to show how our system behaves in a dynamic environment. This clearly is future work. However, we can already say that the only effect of unreachable peers is that advertisement messages and query messages will not arrive. The consequence would be that other peers need to be selected, resulting in an increase of the number of messages and/or a sparser semantic overlay network, both gradually decreasing the performance of our system. We expect that the costs will remain to be low in a dynamic network, because the advertisement process does not consume many messages. This means that restoring the semantic overlay would not have a dramatic effect on the network load.

7.7 Experimental settings

In our experiments, we have systematically simulated various settings with different values of input variables. In the following, we describe an interesting selected subset of the settings to prove the validity of our hypotheses.

7.8 Setting 1

In the first setting, we use a naive peer selection algorithm, which selects *n random* peers from the set of peers that are known from advertisements received, but disregarding the content of the advertisement. This means that peers only have pointers to peers without knowing their expertise, so peer selection would be identical to random selection like in the Gnutella approach. In the experiments, we keep $n = 2$ fixed in every setting, as a rather arbitrary choice. Different values for n yield similar results, but degenerate to a sequence in the case of $n = 1$ and to a broadcast in the case where n is the number of all known peers.

7.9 Setting 2

In the second setting, we apply the expertise-based selection algorithm. The *best n* ($n = 2$) peers are selected for query forwarding. Here, the peer selection algorithm only considers *exact* matches of topics, which means that a peer only is selected when its expertise description contains at least one of the topics from the query abstraction. In this setting, all advertisements are accepted.

7.10 Setting 3

In the third setting, we modify the peer selection algorithm to use the ontology-based similarity measure, instead of only exact matches. The peer selection only

selects peers whose expertise is equally or more similar to the topics from the query abstraction than the expertise of the forwarding peer itself. This method guarantees that queries are forwarded to equal or better experts than the forwarding peer. The danger of this approach is that some of the forwarding branches get stuck in a local maximum because it does only know, if any, peers which are worse matches than itself. In this setting, all advertisements are accepted.

7.11 Setting 4

In the fourth setting, we modify the peer to only accept advertisements that are semantically similar to its own expertise. The threshold for accepting advertisements was set to accept on average half of the incoming advertisements. The peer selection algorithm is identical to the previous setting, namely select peers based on the ontology-based similarity measure.

7.12 Setting 5

In this setting, we assume global knowledge to impose a perfect overlay network on the peer network. In this perfect overlay network, the *knows* relation coincides with the ACM topic hierarchy: Every peer knows exactly those peers that are experts on the neighboring topics of its own expertise. This setting is only applicable for the distribution of the publications according to their topics, as it assumes exactly one expert per topic. A way to achieve this overlay network is via DHT, where for each key (i.e. topic) only one peer is responsible. This means that in this setting we build the semantic overlay on top of the assumed DHT overlay. Clearly, this setting suffers from some limitations as DHT like load-balancing problems in case of popular content, or unreachable content classified on a topic when the peer on the topic does not respond. In this setting, an advertisement is accepted only when the contained expertise description is similar to the receivers own expertise description, thus like in setting 4.

The Table 1 summarizes the instantiations of the input variables for the described settings.

7.13 Results

Figures 2 through 5 show the results for the different settings and distributions. The simulations have been run with a varying number of allowed hops. In the

Table 1 Overview of the simulation settings

Setting no.	Peer selection method	Advertisement method	Topology
Setting 1	Random	Accept all	Random
Setting 2	Exact match	Accept all	Random
Setting 3	Ontology-based match	Accept all	Random
Setting 4	Ontology-based match	Accept similar	Random
Setting 5	Ontology-based match	Accept similar	Perfect

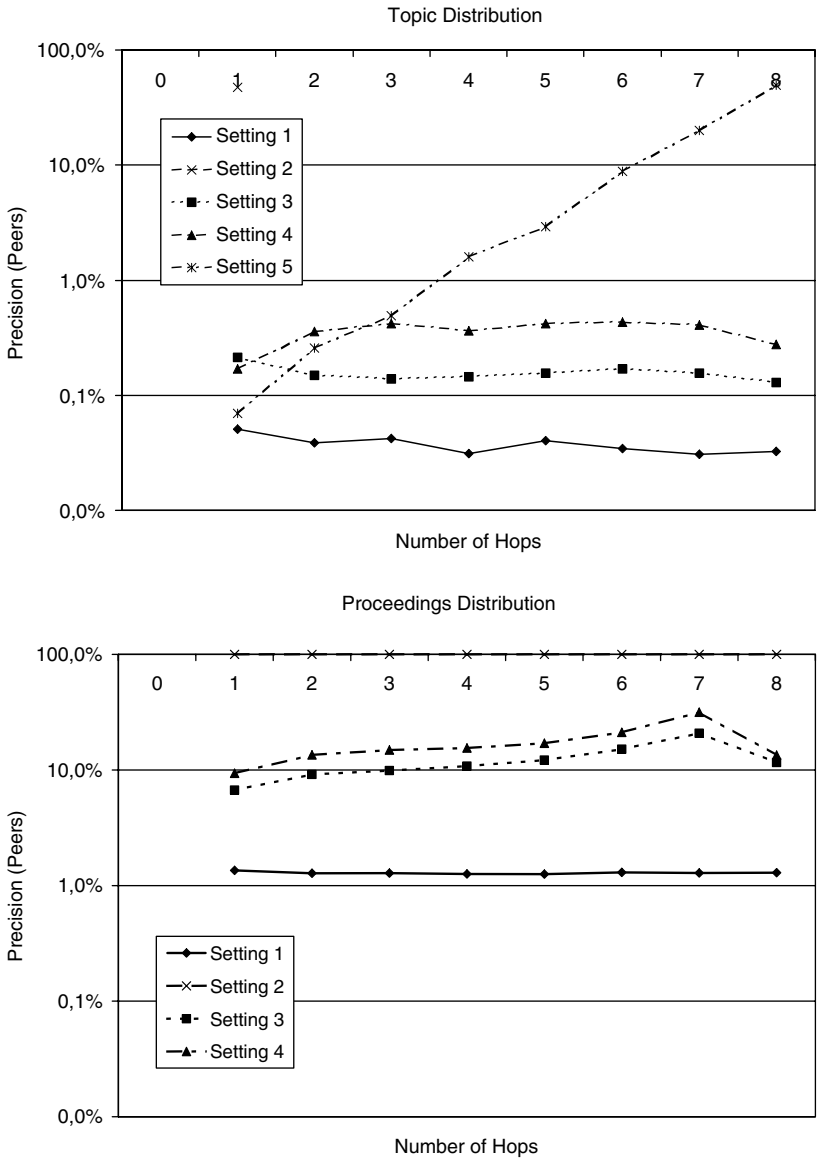


Fig. 2 *Precision*$_{Peers}$

results, we show the performance for a maximum of up to eight hops. Zero hops means that the query is processed locally and not forwarded. Please note that the diagrams for the number of messages per query and recall (i.e. Figs. 3–5) present cumulative values, i.e. they include the sum of the results for *up to n* hops. The diagram for the precision (Fig. 2) of the peer selection displays the precision for a particular number of hops.

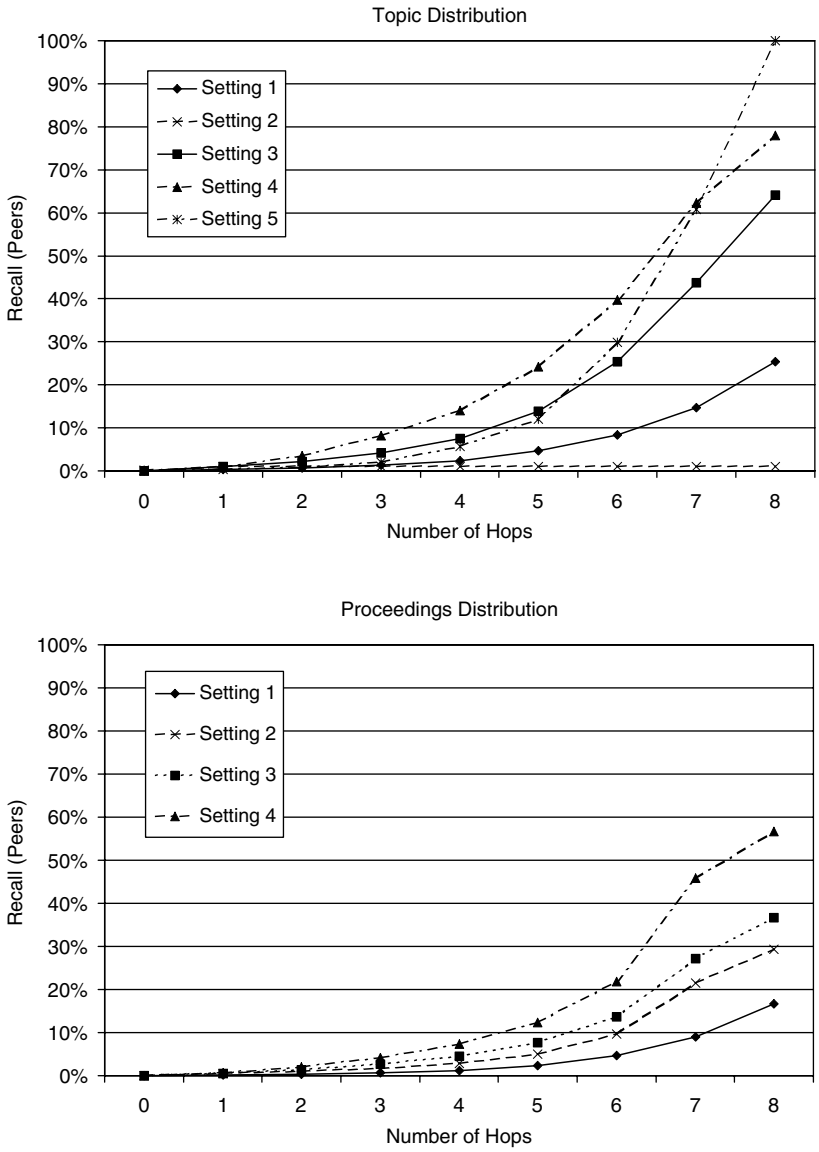


Fig. 3 *Recall*$_{Peers}$

In the following, we interpret the results of the experiments for the various settings described above with respect to our hypotheses H1 through H4.

7.13.1 R1: Expertise-based selection

The results of Fig. 2, Setting 1, show that the naive approach of random peer selection gives a constant low precision of 0.03% for the topic distribution and 1.3%

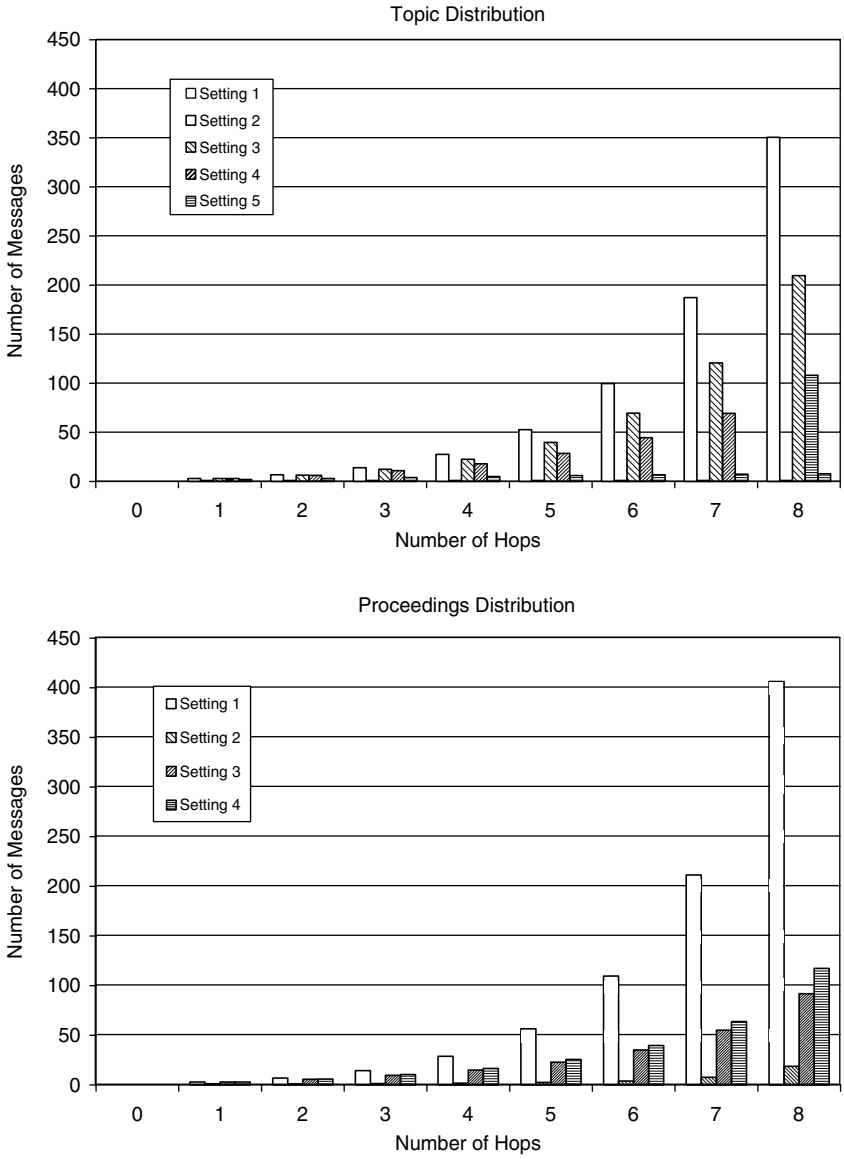


Fig. 4 $Number_{Messages}$

for the proceedings distribution. This results in a fairly low recall of peers and documents despite a high number of messages, as shown in Figs. 3–5, respectively. With the expertise-based selection, either exact or similarity based matching, the precision can be improved considerably by about one order of magnitude. For example, with the expertise-based selection in Setting 3, the precision of the peer selection (Fig. 2) can be improved from 0.03 to 0.15% for the topic distribution and from 1.3 to 15% for the proceedings distribution. With the precision, also the

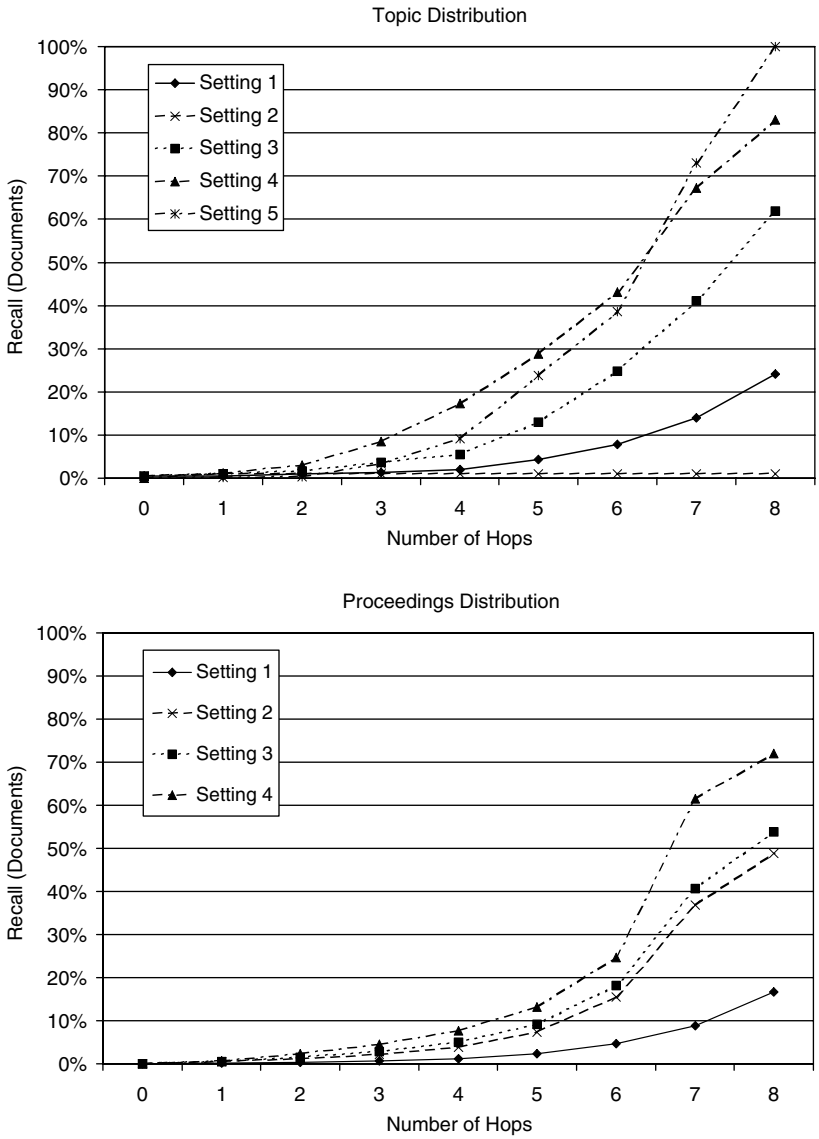


Fig. 5 $Recall_{Documents}$

recall of peers and documents rises (Figs. 3 and 5). At the same time, the number of messages per query can be reduced. The number of messages sent is influenced by two effects. The first effect is message redundancy: The more precise the peer selection, the higher is the chance of a peer receiving a query multiple times on different routes. This redundancy is detected by the receiving peer, which will forward the query only once, thus resulting in a decreasing number of queries sent across the network. The other effect is caused by the selectivity of the peer selection: It only forwards the query to peers whose expertise is semantically more

or equally similar to the query than that of the own expertise. With an increasing number of hops, as the semantic similarity of the expertise of the peer and the query increases, the chance of knowing a qualifying peer decreases, which results in a decrease of messages.

7.13.2 R2: Ontology-based matching

The result of Fig. 2, Setting 2, shows that the exact match approach results in a maximum precision already after one hop, which is obvious because it only selects peers that match exactly with the query's subject. However, Fig. 3 shows that the recall in this case is very low in the case of the topic distribution. This can be explained as follows: For every query subject, there is only one peer that exactly matches in the entire network. In a sparse overlay network, the chance of knowing that relevant peer is very low. Thus, the query cannot spread effectively across the network, resulting in a document recall of only 1%. In contrary, Setting 3 shows that when semantically similar peers are selected, it is possible to improve the recall of peers and documents, to 62% after eight hops. Also in the case of the proceedings distribution, where multiple exact matches are possible, we see an improvement from 49% in the case of exact matches (Setting 2), to 54% in the case of ontology based matches (Setting 3). Naturally, this approach requires to send more messages per query and also results in a lower precision.

7.13.3 R3: Semantic overlay network

In Setting 4, the peers only accept semantically similar advertisements. This has proven to be a simple, but effective way for creating a semantic overlay network that correlates with the expertise of the peers. This allows to forward queries along the gradient of increasing semantic similarity. When we compare this approach with that of Setting 3, the precision of the peer selection can be improved from 0.15 to 0.4% for the topic distribution and from 14 to 20% for the proceedings distribution. The recall of documents can thus be improved from 62 to 83% for the topic distribution and from 54 to 72% for the proceedings distribution.

It is also interesting to note that the precision of the peer selection for the similarity based matching decreases slightly after seven hops (Fig. 2). The reason is that after seven hops the majority of the relevant peers has already been reached. Thus, the chance of finding relevant peers decreases, resulting in a lower precision of the peer selection.

7.13.4 R4: The "perfect" overlay network

The results for Setting 5 show how one could obtain the maximum recall and precision, if it were possible to impose an ideal semantic overlay network. All relevant peers and thus all bibliographic descriptions can be found in a deterministic manner, as the query is simply routed along the route which corresponds to the shortest path in the ACM topic hierarchy. At each hop, the query is forwarded to exactly one peer until the relevant peer is reached. The number of messages required per query is therefore the length of the shortest path from the topic of

expertise of the originating peer to that of the topic of the query subject. The precision of the peer selection increases to the maximum when arriving at the eighth hop, which is the maximum possible length of a shortest path in the ACM topic hierarchy. Accordingly, the maximum number of messages (Fig. 4) required is also eight.

8 The Bibster field experiment

In addition to the simulation experiments, we have evaluated the methods of expertise-based peer selection in a realistic field-experiment, as part of the Bibster system. The Bibster system[5] [11] was developed as part of the EU-funded SWAP project, with contributions by many of the project team. We have implemented the methods for expertise-based peer selection in the Bibster system, and performed a public field experiment to evaluate the model in a real-world setting. We are aware that the data obtained in the field experiment does not allow to make statements about statistical significant. It therefore should be seen as an addition to our simulation results and a case study for a real-life deployment.

The Bibster system Bibster is a Peer-to-Peer system for exchanging bibliographic data among researchers. Bibster exploits ontologies in data storage, query formulation, query routing and answer presentation: When bibliographic entries are made available for use in Bibster, they are structured via the SWRC ontology and classified according to the ACM topic hierarchy, both earlier mentioned in this paper. This ontological structure is then exploited to help users formulate their queries. Subsequently, the ontologies are used to improve query routing across the Peer-to-Peer network. Finally, the ontologies are used to post-process the returned answers in order to do duplicate detection. Bibster is a fully implemented open source solution built on top of the JXTA platform.

8.1 Setup of the field experiment

The Bibster system was made publicly available and advertised to researchers in the Computer Science domain. The evaluation was based on the analysis of system activity that was automatically logged to log files on the individual Bibster clients. In Bibster, two different peer selection algorithms ran at the same time, namely our expertise-based peer selection and a random query forwarding algorithm. We have analyzed the results for a period of three months (June–August 2004).

Three hundred and ninety-eight peers spread across multiple organizations mainly from Europe and North America participated in the field experiment and used the Bibster system.

A total of 98,872 bibliographic entries were shared by the 398 peers, with an average of 248 entries per peer. However, the distribution had a high variance (cf. Fig. 6):

While 62% (248 peers) were free-riding[6] and shared no content, 6% (24 peers) shared at least 1000 entries each, accounting for 79% of the total shared content.

[5] http://bibster.semanticweb.org.

[6] In many Peer-to-Peer systems (e.g. Napster, Gnutella) users are mainly interested in their own advantage and conserve their resources (i.e. bandwidth) by sharing no files. In the common

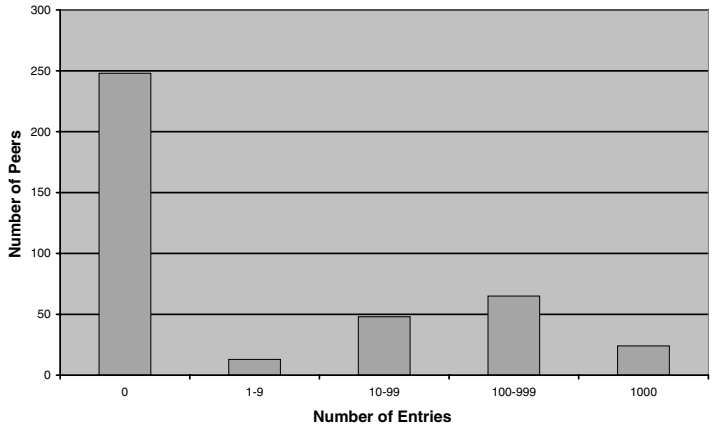


Fig. 6 Distribution of shared content

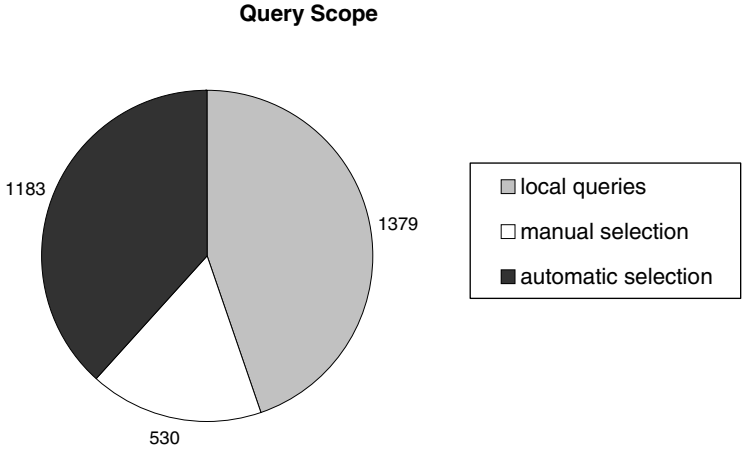


Fig. 7 Scope of queries

With respect to the variance, the distribution is similar to that of the *topic distribution* from the simulation experiments, where many peers provided no entries (those whose topic had no classified instances) and few peers provided many entries (those with popular topics such as "Database Management"). The users performed a total of 3319 queries. With respect to the scope of the queries, Fig. 7 shows that the users mainly performed queries on their local peers and automatic search across the entire network. Only in few cases, the queries were directed to a manually selected peer. This confirms the need for efficient peer selection algorithms. For the 3319 queries, the users received a total of 36,960 result entries, i.e. around 11 result entries per query. Result entries were actively used 801 times, i.e. copied or stored locally.

literature, this phenomena is called *Free-Rider* problem. Users do not have a direct incentive to share files.

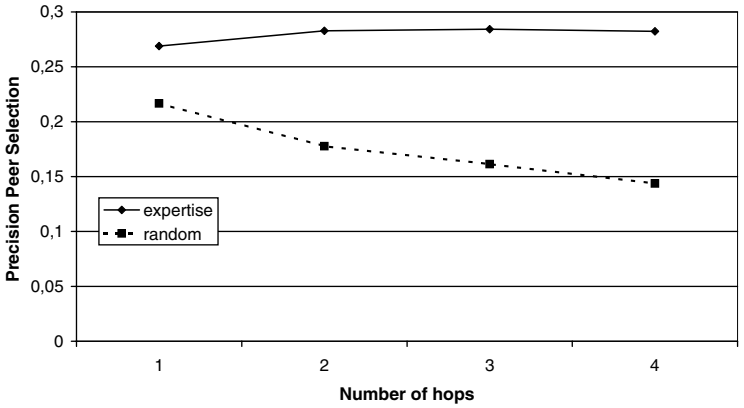


Fig. 8 *Precision*$_{Peers}$

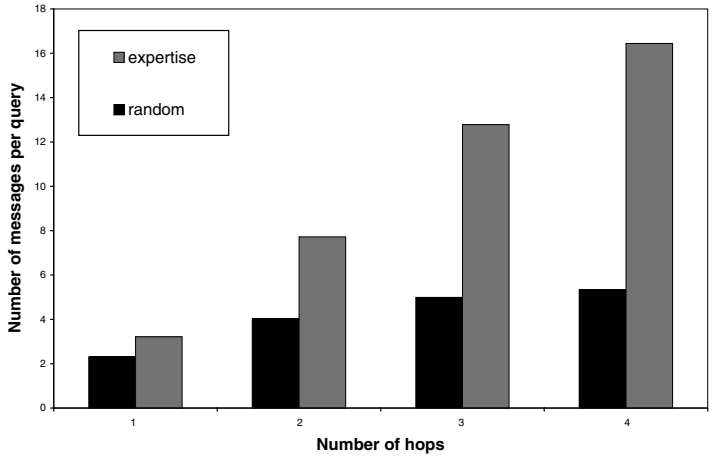


Fig. 9 *Number*$_{Messages}$

8.2 Results

With respect to query routing and the use of the expertise-based peer selection, we were able to reduce the number of query messages by more than 50%, while retaining the same recall of documents compared with a naive broadcasting approach. Figure 8 shows the precision of the peer selection (the percentage of the reached peers that actually provided answers to a given query): While the expertise-based peer selection results in an almost constant high precision of 28%, the naive algorithm results in a lower precision decreasing from 22% after one hop to 14% after four hops.[7]

Figure 9 shows the number of forwarded query messages sent per query. It can be seen that with an increasing number of hops, the number of messages

[7] The decrease is due the redundancy of relevant peers found on different message paths: Only distinct relevant peers are considered.

sent with the expertise-based peer selection is considerably lower than with the naive algorithm. Although we have shown an improvement in the performance, the results also show that with a network of the size as in the field experiment, a naive approach is also acceptable. On the other hand, with a growing number of peers, query routing and peer selection becomes critical. In the previous discussed simulation experiments, networks with thousands of peers improve in the order of one magnitude in terms of recall of documents and relevant peers.

8.3 Comparison with results from simulation experiments

Overall, the results of the simulation experiments have been validated: We were able to improve the precision of the peer selection and thus reduce the number of sent messages. However, the performance gain by using the expertise-based peer selection was not as significant as in the simulation experiments.[8]

This is mainly due to the following reasons:

- *Size of the network* The size of the network in the field experiment was considerably *smaller* than in the simulation experiments. While the total number of participating peers was already fairly large (398), the number of peers online at any point in time was fairly small (order of tens).
- *Network topology* In the field experiment, we built the semantic overlay network on top of the JXTA network. Again, because of the small size of the network, the JXTA topology degenerates to a fully connected graph in most cases. Obviously, for these topologies, a naive algorithm yields acceptable results.
- *Distribution of the content* In the simulation experiments, we distributed the shared content according to certain assumptions (based on topics, conferences, journals). In real-world experiments, the distribution is much more heterogeneous, both in terms of the expertise of the peers and the amount of shared content.

9 Conclusions and future work

In this paper, we have presented a model for expertise-based peer selection, in which a semantic overlay network among the peers is created by advertising the expertise of the peers. We have shown how the model can be applied in a bibliographic scenario. Simulation experiments that we performed with this bibliographic scenario show the following results:

- Using expertise-based peer selection can increase the performance of the peer selection by an order of magnitude (result R1).
- However, if expertise-based peer selection uses simple exact matching, the recall drops to unacceptable levels. It is necessary to use an ontology-based similarity measure as the basis for expertise-based matching (result R2).

[8] In terms of recall, there were no improvements at all, as even the naive algorithm generally was able to reach all relevant peers.

– An advertising strategy where peers only accept advertisements that are semantically close to their own profile (i.e. that are in their semantic neighborhood) is a simple and effective way of creating a semantic overlay network. This semantic overlay network allows to forward queries along the gradient of increasing semantic similarity (result R3).
– The above results depend on how closely the semantic overlay network of the network mirrors the structure of the ontology. All relevant performance measure reach their optimal value when the network is organized exactly according to the structure of the overlay network (result R4). Although this situation is idealized and in will in practice not be achievable, the experiment serves to confirm our intuitions on this.

Also, the field experiment showed that we were able to improve the precision of the peer selection and thus reduce the number of sent messages. However, the performance gained by using the expertise-based peer selection was not as significant as in the simulation experiments. Summarizing, in both the simulation experiments and the field experiments, we have shown that expertise-based peer selection combined with ontology-based matching outperforms both random peer selection and selection based on exact matches, and that this performance increase grows when the semantic topologies more closely mirrors the domain ontology.

We have made a number of simplifying assumptions in our experiments, such as the assumption that all peers agree on the use of a single ontology, which is not realistic in all cases. We already have work in progress which allows us to relax this constraint. We expect that differences in ontologies used by different peers will *lower* our results, since the computation of the semantic distance between peers becomes less reliable across different ontologies. Currently, we are working on an approach where expertise descriptions are not described in terms from a global shared ontology. Instead, routing is based on overlap between sets of locally extracted terms.

In our simulation experiments, the semantic overlay network was determined once, during an initial advertising round, and was not adapted any further during the lifetime of one experiment. In our field experiment, this assumption was not made and also the work in [27] shows how the overlay network can be adjusted based on the exchange of queries and answers. More research has to be done to show that such a self-adjusting network will *improve* the results. We think this will be the case since the semantic overlay network will converge better towards the structure of the underlying ontology than our current one-shot advertising allows. Currently, we submitted a paper containing results on simulations with a network where content is distributed dynamically and peers update and re-advertise their expertise descriptions. In that paper, we also used a more complex expertise models based on Latent Semantic Indexing.

The expertise model presented for the bibliographic scenario used in our simulations experiments is a fairly simple one, based on the ACM topic hierarchy. Other domains may require more complex expertise models with different similarity functions. One option would be, for example, to extend the expertise model with quantitative measures to indicate how much information for a certain topic of expertise is available on the peer. Another option, on which we are currently working, is to automatically extract a shared term similarity matrix based on a subset of documents retrieved from the network.

Acknowledgements Research reported in this paper has been partially financed by the EU in the IST project SWAP (IST-2001-34103). We would like to thank our colleagues for fruitful discussions.

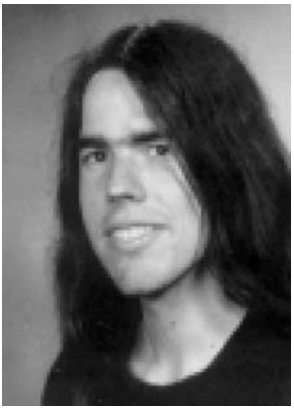
References

1. Aberer K (2001) P-grid: a self-organizing access structure for p2p information systems. In: Proceedings of the sixth international conference on cooperative information systems (CoopIS 2001), vol. 2172 of lecture notes in computer science. Springer Verlag, Trento, Italy. http://www.p-grid.org/Papers/CoopIS2001.pdf
2. Aberer K, Cudré-Mauroux P, Hauswirth M, Pelt TV (2004) Gridvine: building internet-scale semantic overlay networks. In: McIlraith SA, Plexousakis D, van Harmelen F (eds) Proceedings 3rd international semantic web conference (ISWC2004), vol. 3298 of lecture notes in computer science. Springer, Hiroshima, Japan, pp 107–121
3. Ahlborn B, Nejdl W, Siberski W (2002) OAI-P2P: A peer-to-peer network for open archives. In: Proceedings of the 2002 international conference on parallel processing workshops (ICPPW'02). http://www.kbs.uni-hannover.de/Arbeiten/Publikationen/2002/oaip2p.pdf
4. Berners-Lee T, Hendler J, Lassila O (2001) The semantic web. Scientific American
5. Broekstra J, Kampman A (2004) SeRQL: an RDF query and transformation language. Submitted to the International Semantic Web Conference, ISWC 2004. See also http://www.openrdf.org/doc/SeRQLmanual.html
6. Byers J, Considine J, Mitzenmacher M (2002) Simple load balancing for distributed hash tables. Technical report, CS Department, Boston University
7. Clarke I, Sandberg O, Wiley B, Hong T (2001) Freenet: a distributed anonymous information storage and retrieval system. In: Proceedings of the international workshop on design issues in anonymity and unobservability, pp 46–66
8. Crespo A, Garcia-Molina H (2002) Routing indices for peer-to-peer systems. In: ICDCS'02 proceedings of the 22nd international conference on distributed computing systems (ICDCS'02), IEEE Computer Society, Washington, DC, USA, p 23
9. Ehrig M, Schmitz C, Staab S, Tane J, Tempich C (2003) Towards evaluation of peer-to-peer-based distributed knowledge management systems. In: Proceedings of the AAAI spring symposium agent-mediated knowledge management (AMKM-2003)
10. Finin T, Fritzson R, McKay D, McEntire R (1994) KQML as an Agent Communication Language. In: Adam N, Bhargava B, Yesha Y (eds) Proceedings of the 3rd international conference on information and knowledge management (CIKM'94). ACM Press, Gaithersburg, MD, USA, pp 456–463. citeseer.csail.mit.edu/finin94kqml.html
11. Haase P, Broekstra J, Ehrig M, Menken M, Mika P, Plechawski M, Pyszlak P, Schnizler B, Siebes R, Staab S, Tempich C (2004) Bibster—a semantics-based bibliographic peer-to-peer system. In: Proceedings of the 3rd international semantic web conference. Hiroshima, Japan. http://www.aifb.uni-karlsruhe.de/WBS/pha/publications/bibster04iswc.pdf
12. Haase P, Siebes R, van Harmelen F (2004) Peer selection in peer-to-peer networks with semantic topologies. In: Bouzeghoub M (ed) Proceedings of the international conference on semantics in a networked world (ICNSW'04), vol 3226 of LNCS. Springer Verlag, Paris, pp 108–125
13. Jr RJB, Bohrer W, Brice RS, Cichocki A, Fowler J, Helal A, Kashyap V, Ksiezyk T, Martin G, Nodine MH, Rashid M, Rusinkiewicz M, Shea R, Unnikrishnan C, Unruh A, Woelk D (1997) Infosleuth: semantic integration of information in open and dynamic environments (experience paper). In: Peckham J (ed) SIGMOD 1997, Proceedings ACM SIGMOD international conference on management of data, Tucson, Arizona, USA, ACM Press, pp 195–206
14. Kan G (2001) Gnutella. In: Oram A (ed) Peer-to-peer: harnessing the power of disruptive technologies. O'Reilly and Associates, pp 94–122
15. Lassila O, Swick RR (1999) Resource description framework (rdf) model and syntax specification. http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/
16. Leibowitz N, Ripeanu M, Wierzbicki A (2003) Deconstructing the kazaa network. In: Proceedings of the 3rd IEEE workshop on internet applications (WIAPP'03). Santa Clara, CA
17. Li Y, Bandar ZA, McLean D (2003) An approach for measuring semantic similarity between words using multiple information sources. Trans Knowledge Data Eng 15(4):871–882

18. Löser A, Wolpers M, Siberski W, Nejdl W (2003) Efficient data store discovery in a scientific P2P network. In: Ashish N, Goble C (eds) Proceedings of the WS on semantic web technologies for searching and retrieving scientific data. CEUR WS 83. Colocated with the 2. ISWC-03
19. McIlraith S, Son T, Zeng H (2001) Semantic web services. In: IEEE intelligent systems (Special Issue on the Semantic Web), March/April 2001. citeseer.ist.psu.edu/mcilraith01semantic.html
20. Nejdl W, Wolpers M, Siberski W, Schmitz C, Schlosser M, Brunkhorst I, Löser A (2003) Super-peer-based routing and clustering strategies for rdf-based peer-to-peer networks. In: Proceedings of the 12th international world wide web conference, Budapest, Hungary, citeseer.nj.nec.com/nejdl02superpeerbased.html
21. Rada R, Mili H, Bicknell E, Blettner M (1989) Development and application of a metric on semantic nets. IEEE Trans Syst Man Cybernetics 19(1):17–30
22. Ratnasamy S, Francis P, Handley M, Karp R, Shenker S (2001) A scalable content-addressable network. In: Proceedings of ACM SIGCOMM 01
23. Rowstron A, Druschel P (2001) Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In: IFIP/ACM international conference on distributed systems platforms (Middleware), Heidelberg, Germany, pp 329–350. http://research.microsoft.com/ antr/PAST/pastry.pdf
24. Schlosser MT, Sintek M, Decker S, Nejdl W (2002) Hypercup- hypercubes, ontologies, and efficient search on peer-to-peer networks. In: Moro G, Koubarakis M (eds) AP2PC, vol 2530 of lecture notes in computer science, Springer, pp 112–124
25. Stoica I, Morris R, Karger D, Kaashoek MF, Balakrishnan H (2001) Chord: a scalable peer-to-peer lookup service for Internet applications. In: Proceedings of the ACM SIGCOMM 01
26. Tang C, Xu Z, Dwarkadas S (2002) Peer-to-peer information retrieval using self-organizing semantic overlay networks. Technical report, HP Labs. citeseer.ist.psu.edu/tang03peertopeer.html
27. Tempich C, Staab S, Wranik A (2004) Remindin': semantic query routing in peer-to-peer networks based on social metaphors. In: WWW'04: Proceedings of the 13th international conference on World Wide Web. ACM Press, pp 640–649
28. Voulgaris S, Kermarrec A-M, Massoulie L, van Steen M (2004) Exploiting semantic proximity in peer-to-peer content searching. In: Proceedings of the 10th international workshop on future trends in distributed computing systems (FTDCS), Suzhou, China. http://www.cs.vu.nl/ spyros/papers/ftdcs.04.pdf

Author Biographies

Peter Haase is a researcher in the Knowledge Management group at the Institute of Applied Computer Science and Formal Description Methods (AIFB) at the University of Karlsruhe, Germany. He received his diploma in computer science in October 2001 from the University of Rostock, Germany. From 2001 to 2003, he worked at IBM in the Silicon Valley Labs as a software engineer, before joining the AIFB in April 2003. He was a member of the SWAP (Semantic Web and Peer-to-Peer) project and is currently active as a member in the SEKT (Semantically Enabled Knowledge Technologies) project and as a project leader in the NeOn project (Lifecycle Support for Networked Ontologies). His research interests include ontology management in distributed information systems, semantic interoperability, and ontology evolution.



Ronny Siebes received a M.E. degree in Artificial Intelligence from De Vrije Universiteit Amsterdam in 2001. From 2001 to 2005, he worked as a member of the SWAP (Semantic Web and Peer-to-Peer) project and as a Ph.D. student in Knowledge Representation & Reasoning Group of Prof. Dr. Frank van Harmelen also at the Vrije Universiteit Amsterdam and defended his thesis in June 2006 with the topic "Semantic Routing in Peer-to-Peer Systems". In 2005, he also worked for the Dutch "Multimedian" project, on the development of a set of e-culture demonstrators providing multimedia access to distributed collections of cultural heritage. Currently, he works as a postdoc in the European funded "Open Knowledge" project, on Peer-to-Peer4 service discovery. His research interests include large scale Peer-to-Peer systems, Data Semantics, and Trust Metrics.

Frank van Harmelen is a professor in Knowledge Representation & Reasoning in the AI department (Faculty of Science) at the Vrije Universiteit Amsterdam. After studying mathematics and computer science in Amsterdam, he moved to the Department of AI in Edinburgh, where he was awarded a Ph.D. in 1989 for his research on meta-level reasoning. While in Edinburgh, he worked with Dr. Peter Jackson on Socrates, a logic-based toolkit for expert systems, and with Prof. Alan Bundy on proof planning for inductive theorem proving. After his Ph.D. research, he moved back to Amsterdam where he worked from 1990 to 1995 in the SWI Department under Prof. Wielinga. He was involved in the REFLECT project on the use of reflection in expert systems, and in the KADS project, where he contributed to the development of the $(ML)^2$ language for formally specifying Knowledge-Based Systems. In 1995, he joined the AI research group at the Vrije Universiteit Amsterdam, where he was appointed professor in 2002, and is leading the Knowledge Representation and Reasoning Group. His current interests include Approximate reasoning, Semantic Web and Medical Protocols. He has published three books (on meta-level inference, on knowledge-based systems, and on the Semantic Web) and over 100 research papers, most of which can be found on-line.