

# PROUTE: EXPERTISE-BASED SELECTION USING SHARED TERM SIMILARITY MATRICES

Ronny Siebes                      Spyros Kotoulas

*Department of Computer Science, Vrije Universiteit Amsterdam, The Netherlands  
Ronny@cs.vu.nl*

## Abstract

Finding the right content or experts in a Peer-to-Peer system or Multi-Agent system is a challenging problem especially when it has to be scalable to large networks. The focus of this paper is on distributed search based on using semantic knowledge about the peers in the network. In this paper we present the pRoute system where each peer advertises a short description of its content that it shares that serves as its *expertise description*. Peers remember the advertisements of related peers and thereby form a semantic overlay by which we mean that peers with similar content are grouped together. Peers calculate the similarity between their content descriptions by a term similarity function which, in the ideal case, is identical for all peers. In simulation experiments we compare the performance of different advertisement- and forwarding policies with respect to precision, recall and the number of messages. Simulation results indicate that precision and recall increases when the policies take semantics into account, without an increase of the number of advertisement- and query messages.

## 1 Introduction

Undisclosed content, lack of privacy and the possibility to censor data are seen as important disadvantages of the centralized approach of today's popular search engines. Peer-to-Peer systems, where nobody is in control, are in principal much more difficult to be used for tracing the behavior of users. A big advantage of centralized search engines is that the number of messages needed in the query process often is only 2 and the number of hops is only 1, guaranteeing efficient bandwidth usage and quick response times. In Peer-to-Peer systems, the number of messages and hops mainly depends on how quickly the relevant peers are found.

In this paper we present the pRoute system where we adopt the model of expertise-based peer selection using a shared data-model as is described in the work of Haase et al. [4]. In the expertise-based model, peers (or agents) summarize their content or knowledge in so-called *expertise-descriptions*, being a set of terms provided the shared data-model. The difference with the approach of Haase et al. is that our approach does not use an human-built ontology as the shared data-structure but an automatically generated similarity matrix that provides the semantic distances between terms. The expertise descriptions are spread through the network via *advertisement messages*. In this way, peers become aware about the expertise of other peers, enabling them to route queries only to those peers of which the expertise is semantically close related to the content of the queries.

There is much related work in the area of searching content and experts in P2P networks and can mainly be divided into four approaches. Firstly, there are the broadcast-based approaches like Gnutella<sup>1</sup> and Hypercup [9]. In these systems messages are sent to all, or a random subset of the peers in the network. The approach is very robust and has low network maintenance costs. However, it leads in some cases to an unacceptable large amount of forwarded query messages. Secondly, there are the semi-P2P approaches where some parts are centralized, like the centralized registers in Napster or where there is an hierarchical organization of the network structure where the peers with more resources (e.g. bandwidth, cpu-cycles, memory) get more responsibility in the network like in KaZaa. Also for them, the same disadvantages hold as for pure centralized solution. Thirdly, there are the systems where content is distributed over the network based on Distributed Hash Tables [1, 7, 10, 8]. DHT's are based on a very nice trick to route content (or a pointer to

---

<sup>1</sup>The Gnutella protocol specification v0.4, 2000

the content) to the peer where its 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, URL's or words. The characteristic of this technique is that it allows to route content and queries in  $\log(N)$  steps (where  $N$  is the number of peers in the network) to the right peers. A disadvantage of most DHT approaches is that they have high maintenance costs, due to the during changes in the overlay network as a result of peers joining and leaving. Another disadvantage is that a peer which is responsible for a certain key (for example the hash-key of the string 'Brittney Spears') can become a bottleneck. Fourthly, there is the work on Semantic Overlay Networks, where peers have pointers to other peers that have similar content as themselves. Gridvine [2] uses the semantic overlay for managing and mapping data and meta-data schemas, on top of a physical layer consisting of a DHT-based Peer-to-Peer overlay network. Their disadvantage is the strong dependence on DHT and therefor inherits its specific weaknesses. In pSearch [11] documents are distributed through the P2P network based on their mapping into a shared topic vector. The disadvantage of their approach is the high maintenance costs when the vector has to be changed. In Bibster [4], peers describe their content in terms occurring in a shared topic-hierarchy. The problem there is that the hierarchies in their experiments are made manually by humans. Our pRoute system that we describe in the next section, is also a SON approach similar to Bibster with the difference that in pRoute the shared data-structure is automatically created.

In the next section we describe our pRoute system. Section three is about our experimental setup where the data set and the simulation platform is described. Section four shows the simulation results and section five summarizes our work.

## 2 pRoute: Expertise-Based Peer Selection based on a Shared Similarity Matrix

In this section we explain our expertise-based selection method by showing the individual building blocks of the system and the describing the different advertisement- and query policies.

### 2.1 Elements

- *Neighbor set* Initially each peer has a bootstrap neighbor list in a small, fixed-sized cache of entries (with typical value 20, 50, or 100 slots). A cache entry contains the network address (i.e., IP-address and port) of another peer in the overlay. This set could, for example, be downloaded from a gateway peer, a web-site or come with the download of the implementation.
- *Expertise Description* Each peer has an expertise extraction function  $\epsilon : docs_p \mapsto \mathcal{T}_p$  which maps the content of the documents of the peer  $docs_p$  into a set of terms that occur in the shared term similarity matrix  $\mathcal{T}_p \subset 2^{\text{STRING}}$ , which will be discussed later in this document.
- *Query Abstraction* For simplicity reasons we assume that a query posted by a user is just a set of words, thus without boolean operators or other constructs except an implicit AND between the words. An answer on the query is therefor correct when all the words are in each of the result documents. The words in the query are mapped by a mapping function  $\pi : query \mapsto \mathcal{T}_q$  into a set of terms  $\mathcal{T}_q \subset 2^{\text{STRING}}$  that occur in the shared term similarity matrix.
- *Term Similarity Matrix* A term similarity matrix is a matrix  $\mathcal{S}_{dom} = |\mathcal{T}_{dom}| \times |\mathcal{T}_{dom}|$  about a certain domain  $dom$  which contains semantic distances  $[0, 1]$  between the set of (popular) terms  $\mathcal{T}_{dom}$  in  $dom$ . In this way a peer is able to extract between two terms  $t_1$  and  $t_2$  the semantic similarity  $sim : (t_1, t_2)_{dom} \mapsto [0, 1]$  used by the similarity function which will be described in the next paragraph. A user could download the domain matrices of interest (e.g. computer science or biology) from a web-site or develop one by its own although it is recommended to re-use similarity matrices as much as possible (will be discussed later). There are numerous ways to determine the semantic distance between terms. One way is to do it manually namely by letting a group of domain experts giving the distances. Another way is to use an ontology [4] or another kind of semantic graph where the minimal path distance between topics is an indication of similarity.

In our approach we take a representative subset of documents for a certain domain which we discuss later in the section about the experimental setup. For each document we use a term extraction algorithm developed by natural language processing community to extract a set of terms that describe

the content of the documents. From this way we create a  $Term \times Document$  matrix, where each cell gives the frequency of the term in the corresponding document normalized over all documents. We use Latent Semantic Indexing [3] to automatically transpose the term-by-document matrix into a term-by-term matrix where each cell in the matrix gives a value that represents the semantic similarity between the corresponding terms. It is beyond the scope of this paper to discuss this approach in more detail.

- **Similarity Function** According to the expertise-based selection method, each peer has the complete autonomy to choose its own similarity measure between two expertise descriptions or between an expertise description and a query:

$$\sigma : (\mathcal{T}_e, \mathcal{T}_{eq}) \mapsto [0, 1]$$

where  $\mathcal{T}_e$  is a set of terms from an expertise description and  $\mathcal{T}_{eq}$  a set of expertise terms or abstracted query terms. In our simulations we assume that all peers in the network share documents about the same domain  $dom$  and therefore use the same term similarity matrix from for determining the semantic similarity between terms. Future work has to give an answer on what effect is on letting peers having different distance matrices from different domains and variations of them within the same domain. We instantiate  $\sigma$  in the following way:

$$\sigma(\mathcal{T}_e, \mathcal{T}_{eq}) = \frac{1}{|\mathcal{T}_e|} \sum_{t_i \in \mathcal{T}_e} \max_{t_j \in \mathcal{T}_{eq}} (sim(t_i t_j)_{dom})$$

which means taking the sum of the maxima of term similarities and divide it by the length of the expertise description  $\mathcal{T}_e$ . Note that this function is intensionally asymmetric,  $\sigma(\mathcal{T}_1, \mathcal{T}_2) \neq \sigma(\mathcal{T}_2, \mathcal{T}_1)$ , under the assumption that a user looking for e.g. an expert on Thai food would be satisfied by an expert on both Thai and Chinese, but one looking for an expert that combined French and Japanese would not necessarily be happy with a restaurant that focused on one or the other.

## 2.2 Policies

In this subsection, we describe the policies that concern how queries and advertisements are handled in pRoute. For each policy we give instantiations that we tested in our simulations. More precisely, for each policy we provide one instantiation based on a random selection method and one that uses the shared similarity matrix in the selection process. This allows us to see when and in which degree using the shared similarity matrix is beneficial. For all policies it holds that when a message is sent to a peer that is off-line, the sender will select a new peer from the neighbor list. This means that in a very dynamic system and/or a network where not many peers are on-line, a significant number of messages will be lost. We record this in our statistics and just count them as sent messages.

**Query forwarding policy** Besides that a peer tries to answer a query, peers also select peers to forward the query via the query forwarding policy. This policy is applied when a peer receives a forwarded query, or when a query is initialized by the peers user. We simulated the following instantiations:

- $QF1_{(h,n)}$ : Queries are maximally forwarded  $h$  hops, each step to maximally  $n$  random neighbors.
- $QF2_{(h,n)}$ : Queries are maximally forwarded  $h$  hops, each step to maximally  $n$  neighbors where peers of which the expertise descriptions are semantically closest to the terms in the query are chosen first.

**Advertisement interval policy** This policy regulates at which moment the peer starts to advertise its expertise.

- $AI1_{(r)}$ : Advertising is periodically triggered after  $r$  expertise description updates.
- $AI2_{(s)}$ : Advertising is only triggered when the similarity (cf. formula on similarity) between expertise description from the last advertisement round and its new expertise description is lower than a certain threshold  $s$ .

**Advertisement distribution policy** When the previous policy decides to advertise or when a peer receives an advertisement that it perhaps wants to forward, the advertisement distribution policy determines to whom to send the advertisements. This policy thus both applies to peers that want to advertise their own expertise and to peers that want to forward expertise description other peers.

- $AD1_{(h,n)}$  : In the first hop, advertisements are sent to all neighbors and from the second hop, advertisements are distributed to a random subset of  $n$  neighbors until a maximum of  $h$  hops.
- $AD2_{(h,n,t)}$  : In the first hop, advertisements are sent to those peers of which the expertise description has a similarity to the query above a threshold  $t$ . For two hops and more, the advertisements are sent to the  $n$  peers of which the expertise descriptions are semantically closest to the expertise description until a maximum of  $h$  hops.

**Advertisement acceptance policy** When a peer receives an advertisement it can decide to keep it or to ignore it. In the situation when the maximum number of  $n$  advertisements that a peer wants to store is not reached, we decide either to store every advertisement or the semantic close ones till the maximum is reached. After that, it can be the case that a new advertisement replaces an advertisement that exists in the receivers storage.

We tested the following two instantiations:

- $AA1_{(n)}$ : Only those advertisements are stored of which the expertise descriptions are semantically closest to the receivers own expertise description. This means that when the new advertisements description is closer than the worst description in the storage, it replaces this description.
- $AA2_{(n)}$ : A received advertisement randomly replaces one of the stored advertisements.

### 3 Experimental Setup

We wrote a simulation platform to test the performance of the different policies. We simulate a bibliographic scenario where researchers are represented by peers and share their publications with other researchers via a P2P network. The expertise description of a peer contains the research terms of the researcher it represents, which reflects the content of the documents it shares.

**Peer set, document description set** The document set contains around 100.000 scientific documents in the computer science domain, namely a subset of the DBLP-database [5]. The document crawl procedure is based on a breadth-first search over co-authors, starting by one author. More precisely, we started by crawling and storing all documents and co-authors from the author 'Maarten van Steen'. After that, for all the co-authors we crawled their documents and their co-authors and removed the doubles. This procedure is continued till a given maximum that we put on 80.000 peers. We used an NLP tool called 'TextToOnto' [6] to extract for each document a set of descriptive terms (around 20 terms per document), which resulted in 160K unique terms.

**Shared similarity matrix** We used a random subset of 10K out of the 100K document descriptions from the previous paragraph as a source for creating this similarity matrix. We define a term as a *shared term*, when it is shared by at least three peers (an arbitrary chosen number). This resulted in a set of 25K shared terms. We use the method as described in the previous chapter to calculate the Term  $\times$  Term matrix, based on these 10K document vectors.

**Expertise descriptions** As said in previous sections, an expertise description describes the 'expertise' of a peer by a set of terms from the shared distance matrix. The expertise descriptions in our experiments are made by taking the union of all document descriptions of the peer and filter out the terms which are not in the distance matrix.

Nr	Simulation setting				$Peer_{prec}$	$Doc_{rec}$	$Q_{msg}$	$A_{msg}$
1	$QF1_{(4,3)}$	$AI1_{(1)}$	$AD1_{(3,5)}$	$AA1_{(80)}$	0.0072	0.0281	124	1099
2	$QF2_{(4,3)}$	$AI2_{(0.8)}$	$AD1_{(3,5)}$	$AA1_{(80)}$	0.0594	0.2181	110	959
3	$QF2_{(4,3)}$	$AI1_{(1)}$	$AD2_{(3,5,0.2)}$	$AA1_{(80)}$	0.1052	0.4036	109	401
4	$QF2_{(4,3)}$	$AI1_{(1)}$	$AD1_{(3,5)}$	$AA2_{(80)}$	0.1792	0.3583	64	1017
5	$QF2_{(4,3)}$	$AI1_{(1)}$	$AD2_{(3,5,0.2)}$	$AA2_{(80)}$	0.1308	0.4383	98	459
6	$QF2_{(4,3)}$	$AI2_{(0.8)}$	$AD1_{(3,5)}$	$AA2_{(80)}$	0.1592	0.4003	78	890
7	$QF2_{(4,3)}$	$AI2_{(0.8)}$	$AD2_{(3,5,0.2)}$	$AA2_{(80)}$	0.1203	0.4215	100	398

Table 1: Effect of different combinations of strategies.

**Query abstractions** We create a query abstraction for a peer  $p$  by taking a random set of 3-5 terms from a randomly chosen document description  $d_{desc}$  for a document  $d$  of which the user of  $p$  is the author.

We believe that the creation of this data-set in itself is a contribution of this paper. The data-set is available from the author on request.

We evaluate the different policies and the influence of other system parameters on four criteria: (1) the peer precision  $Peer_{prec}$ , being the ratio between returned peers with matching documents and all returned peers. (2) the document recall  $Doc_{rec}$ , being the ratio between the number of found documents and the matching documents in the network for the given queries. (3)  $A_{msg}$ : the average number of advertisement messages sent per peer during the simulation. (4)  $Q_{msg}$ : the average number of messages used to resolve a query sent by a peer.

## 4 Results

In this section we only show the most interesting results. We are currently working on a journal version where we discuss everything in much more detail.

**Combining different policies** Table 1 shows how the system performs for different combinations of policies. When we compare result 1 with 2-7, we can see that it is beneficial to use for each policy the semantics-based version. The results indicate that semantics-based query forwarding combined with semantics-based advertisement forwarding are the largest contributors for the improvement in recall and reduction of the number of advertisement messages.

**Influence of the number of neighbors** Table 2 shows the effect of varying the number of expertise descriptions that a peer can store. As can be seen, an increase improves the recall and precision, without an increase in the number of query messages although with an increase of the number of advertisements. This increase is because the number of neighbors selected at the first advertisement hop depends on the number of neighbors in the storage: the more neighbors, the bigger the chance that peers are above the similarity threshold. The trade-off between recall and the number of messages is of course dependent on the requirements of the user of the system. However, the results indicate that the positive effect on the recall by sending more advertisement messages in combination with a larger neighbor storage starts to diminish after 60 neighbors (compare experiment 42 with 43). Probably this result is dependent on the number of peers in the network, where the trade-off in larger networks lies at a higher number of peers.

## 5 Conclusions

In this paper we presented the pRoute system where the nodes in a Peer-to-Peer network describe their content in terms occurring in a shared similarity matrix. The peers share their content descriptions with other peers where peers remember only those peers that are relevant (i.e. semantically close) to their own content. In this way, peers form a semantic overlay network. We have shown how the model can be applied in a bibliographic scenario based on a realistic data-set. Simulation experiments that we performed with this

Nr	Simulation setting				$Peer_{prec}$	$Doc_{rec}$	$Q_{msg}$	$A_{msg}$
39	$QF2_{(4,3)}$	$AI2_{(0.8)}$	$AD2_{(3,5,0.2)}$	$AA2_{(10)}$	0.0359	0.1029	72	81
40	$QF2_{(4,3)}$	$AI2_{(0.8)}$	$AD2_{(3,5,0.2)}$	$AA2_{(20)}$	0.0701	0.2400	90	161
41	$QF2_{(4,3)}$	$AI2_{(0.8)}$	$AD2_{(3,5,0.2)}$	$AA2_{(40)}$	0.0995	0.3665	99	274
42	$QF2_{(4,3)}$	$AI2_{(0.8)}$	$AD2_{(3,5,0.2)}$	$AA2_{(60)}$	0.1135	0.4077	99	350
43	$QF2_{(4,3)}$	$AI2_{(0.8)}$	$AD2_{(3,5,0.2)}$	$AA2_{(160)}$	0.1256	0.4419	100	436

Table 2: Influence of number of neighbors

bibliographic scenario where all peers share the same distance matrix show that it performs much better in recall and the reduction of the number of messages compared to a random approach. Our results are based on a large data-set and are comparable with the work of [4], except that our shared data-structure is richer and created automatically.

## 6 Acknowledgments

We thank Frank van Harmelen for his very useful comments and suggestions. We thank Spyros Voulgaris for his help to crawl the dataset that we used for our experiments. The work was partially supported by the the European funded IST project called 'SWAP' (IST-2001-34103).

## References

- [1] Karl Aberer, Philippe Cudré-Mauroux, Anwitaman Datta, Zoran Despotovic, Manfred Hauswirth, Magdalena Puceva, and Roman Schmidt. P-grid: a self-organizing structured p2p system. *SIGMOD Rec.*, 32(3):29–33, 2003.
- [2] Karl Aberer, Philippe Cudré-Mauroux, Manfred Hauswirth, and Tim Van Pelt. Gridvine: Building internet-scale semantic overlay networks. In *3rd International Semantic Web Conference (ISWC2004)*, pages 107–121, Hiroshima, Japan, 7-11 November 2004.
- [3] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.
- [4] P. Haase, R. Siebes, and F. van Harmelen. Peer selection in peer-to-peer networks with semantic topologies. In Mokrane Bouzeghoub, editor, *Proceedings of the International Conference on Semantics in a Networked World (ICNSW'04)*, volume 3226 of *LNCS*, pages 108–125, Paris, June 2004. Springer Verlag.
- [5] Michael Ley. DBLP Bibliography.  
<http://dblp.uni-trier.de/>.
- [6] Alexander Maedche and Steffen Staab. Ontology learning for the semantic web. *IEEE Intelligent Systems*, 16(2):72–79, 2001.
- [7] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker. A scalable content-addressable network. In *Proceedings of ACM SIGCOMM '01*, 2001.
- [8] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, pages 329–350, Heidelberg, Germany, November 2001.
- [9] Mario T. Schlosser, Michael Sintek, Stefan Decker, and Wolfgang Nejdl. Hypercup - hypercubes, ontologies, and efficient search on peer-to-peer networks. In *AP2PC*, pages 112–124, 2002.

- [10] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for Internet applications. In *Proceedings of the ACM SIGCOMM '01*, 2001.
- [11] C. Tang, Z. Xu, and S. Dwarkadas. Peer-to-peer information retrieval using self-organizing semantic overlay networks. Technical report, HP Labs, November 2002.