

RS2D: Fast Adaptive Search for Semantic Web Services in Unstructured P2P Networks

Ulrich Basters and Matthias Klusch

German Research Center for Artificial Intelligence
Stuhlsatzenhausweg 3, 66121 Saarbruecken, Germany
`uli@basters.de`, `klusch@dfki.de`

Abstract. In this paper, we present an approach, called RS2D v1, to adaptive probabilistic search for semantic web services in unstructured P2P networks. Each service agent dynamically learns the averaged query-answer behavior of its neighbor peers, and forwards service requests to those with minimal mixed Bayesian risk of doing so in terms of estimated semantic gain and communication cost. Experimental evaluation shows that the RS2D search mechanism is robust against changes in the network, and fast with reasonably high precision compared to other existing relevant approaches¹.

1 Introduction

Agent based service provision in the future open semantic Web in one extreme would ad hoc connected autonomous agents require to efficiently search for relevant services without any central coordination means and prior knowledge about their environment. This corresponds to the known challenge of searching unstructured peer-to-peer (P2P) networks with reasonably high performance and low communication efforts, but no prior knowledge about service distribution, ontologies, or network topology. Different solutions to this problem have been proposed in the P2P research literature; an accessible survey is provided in [1]. In contrast to structured P2P networks, unstructured approaches lack of global routing guarantees provided by the overlay, that is, they offer arbitrary network topology, file placement and search. Though blind flooding based search, or variants of it, in such networks like Gnutella [9], performs very robustly with high precision but suffers from insufficient scalability due to high communication overhead. Randomized routing usually keeps the communication effort low, but exhibits low performance due to its random nature and inability to adapt to different query loads. Approaches to informed probabilistic adaptive P2P search in unstructured P2P networks like in [15] improve on such random walks but do not exploit the qualitative results of semantic matching of services to drive the search. In fact, as of today, there exist only few approaches that explicitly perform semantic service retrieval in unstructured P2P networks, that are

¹ This work has been supported by the German ministry of education and research (BMB+F 01-IW-D02-SCALLOPS), and the European Commission (FP6 IST-511632-CASCOM).

Bibster[10] and GSD[5]. Both approaches are still limited in the sense that they require prior knowledge on the semantic overlay, and a fixed (static) global ontology to be used by each peer, respectively.

This motivated us to develop the first risk-driven and behavior based approach, named RS2D, to combine adaptive probabilistic with semantic search for OWL-S services in unstructured P2P networks. RS2D enables each autonomous peer to dynamically build up and manage its own local ontology in OWL based on a minimal shared vocabulary only. Furthermore, and more importantly, RS2D peers do not perform object-specific but average peer behavior based semantic search, that is each of them quickly learns which of its direct neighbours in the network will most probably return relevant semantic web services for a given query with minimal mixed Bayesian risk of both semantic loss and high communication in average. Although the implemented RS2D routing decision scheme works for OWL-S services only, its core decision routing mechanism is independent from the service description language, hence can be, in principle, used to search for any other kind of web services as well, such as WSDL, WSDL-S, and WSMO, by means of replacing our OWLS-MX matchmaker at each peer with an adequate one for local semantic service selection.

The remainder of this paper is organized as follows. Section 2 presents the original version of our RS2D approach (RS2D v1) for risk driven probabilistic search of OWL-S services in unstructured P2P service networks in detail. Sections 3 and 4 provide the results of the comparative experimental evaluation of RS2D v1 and related work, respectively, and we conclude in section 5. Complementary work on RS2D v2 with alternate semantic loss function is in [4].

2 Adaptive Search for OWL-S Services

How does the adaptive probabilistic search for OWL-S services in unstructured P2P networks according to the RS2D routing decision scheme? Each peer first determines the set of OWL-S services that are semantically relevant to a given service request (query) by use of its local OWL-S matchmaker, and then forwards the same query only to those of its direct neighbor peers from which it expects to most probably receive semantically relevant services in reasonable time. For this purpose, it continuously observes the individual average query answering behavior of these peers, caching not only the number but semantic qualities of their replies, and requests in its local training set and updates. These data are then used to estimate the mixed conditional probabilistic risk of semantic query routing for each of its neighbor peers. We present the details of this adaptive mixed risk-driven search in subsequent sections.

2.1 Local Observations

What does an RS2D enabled peer v in an unstructured network observe in concrete terms? From each reply to a given query q it receives from some of its direct neighbour v_k , it extracts data into a training record

$$t = (q, S_q, S'_q, L(S_q), L(S'_q), fid, tid, c_j, a)$$

and stores it in its local training set TS . These observation data are as follows:

- q : Request in terms of the description of a desired service written in a semantic web service description language such as OWL-S.
- S_q : Set of top- k relevant services retrieved *before* forwarding the request.
- S'_q : Set of top- k relevant services retrieved *after* forwarding the request.
- $L(S_q), L(S'_q)$: Semantic score of S_q, S'_q . This score measures the semantic quality of the set of retrieved services with respect to q by summarizing the numeric results of their semantic matching performed by an appropriate matchmaker, that is $L(S_q) := \sum_{s \in S_q} \sigma(s, q)$. For example, searching for OWL-S services in RS2D enabled P2P networks may be achieved by using the OWLS-MX matchmaker (cf. section 2.2), as we did for evaluating the RS2D performance (cf. section 4).
- fid, tid : Identifier of the peer from/to which the request was received/forwarded.
- c_j : Query answer result class, that is c_0 meaning that the query got rejected because it was processed by v_k already, or c_1 meaning that v_k answers to the query with a semantic gain, i.e. with $L(S'_q) - L(S_q) > 0$. That is, the reply is considered of real benefit for v , only if it contains services previously unknown to v with higher degree of semantic relevance.
- a : Communication effort entailed by the decision to route the request to v_k , i.e. the number of message hops in the routing subtree of the request.

The observation vector $x \in \mathbb{N}^2$ used for risk estimations is defined as $x = (fid, tid)$. Our experiments showed, that already the use of these two parameters yield an reasonably well prediction. To be able to predict the values of $\lambda, E(y), E(a)$ and $P(c_j|x)$, we filter the training set TS in different ways.

2.2 Local Semantic Service Selection

In order to compute the numeric semantic scores $LS(S_q)$, each RS2D peer maps the output of its local OWL-S service matchmaker OWLS-MX [12] to the interval $[0, 1]$ (cf. figure 1). The mapping is defined by $\sigma_{mx} : (S^* \times S^*) \mapsto [0, 1] \subset \mathbb{R}$; $\sigma_{mx}(s, q) := (5 - dom + sim)/6$ with $dom = \{0, 1, 2, 3, 4, 5\}$ coding the degrees of match (exact = 0, plug-in = 1, subsumes = 2, subsumed-by = 3, nearest-neighbour = 4, fail = 5), and $sim \in [0, 1]$ the syntactic similarity value, as returned by the matchmaker OWLS-MX.

OWLS-MX takes any OWL-S service description as a query, and returns an ordered set of relevant services that match the query in terms of both crisp logic based and syntactic similarity. For this purpose, it applies five different hybrid matching filters with one selected IR similarity metric each. Logical subsumption failures produced by the integrated OWL-DL reasoner Pellet are tolerated, if the computed syntactic similarity value is sufficient. It turned out that this is beneficial in terms of both recall and precision. For more details on OWLS-MX, we refer the interested reader to [12]; the sources are available at [13].

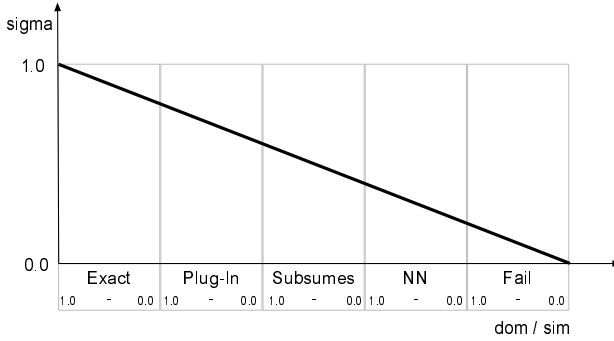


Fig. 1. Mapping of the degrees of semantic service matching returned by the OWLS-MX matchmaker to $[0,1]$ used by RS2D peers to determine the numeric semantic gain of received replies from their neighbour peers

What makes OWLS-MX particularly suitable to semantic service selection in unstructured P2P service networks is its ability to dynamically maintain a local (matchmaker) ontology on the basis of a shared minimal vocabulary of primitive components only. This renders RS2D peers independent from the exclusive use of a fixed global ontology. Instead the minimal set of shared primitive components from which each peer may canonically define local concept semantics constitutes the weakest form of a shared ontology. Further, we assume that RS2D peers do exchange not only the names but terminologic expansions of service I/O concepts. This enables peers to gradually learn about the expertise of other peers through appropriate classification of the concepts of their queries and replies into its local ontology. The same holds in case service descriptions change. Hence, RS2D peers do not have to adopt a static global ontology like in GSD[5].

2.3 Individual Semantic Loss

The estimated semantic loss of routing some query q to a peer v_k (alternatives α_0, α_1) for possible query answer classes (c_0, c_1) based on its average Q/A behavior according to the actual training set is essentially driving the risk based decision making of each RS2D peer. In the original version of the RS2D system v1, the semantic loss $\lambda(\alpha_i, c_j)$ is computed as follows:

$$\begin{array}{c|c|c}
 & \lambda(\alpha_0|\cdot) & \lambda(\alpha_1|\cdot) \\
 \hline
 c_0 & 0 & 2\kappa \\
 c_1 & E(y) - E(a)\kappa & -E(y) + E(a)\kappa
 \end{array} \quad (1)$$

The average message transmission costs are denoted by κ , and assumed to be constant. In addition, the average expected semantic gain $E(y)$ and average number of messages $E(a)$ are defined as follows:

$$E(y) := \frac{1}{|TS_{fid,tid}|} \sum_{t \in TS_{fid,tid}} [L(S''_q)]_t - [L(S'_q)]_t \quad (2)$$

$$E(a) := \frac{1}{|TS_{fid,tid}|} \sum_{t \in TS_{fid,tid}} [a]_t \quad (3)$$

with $[x]_t$ extracting the parameter x from observation record t in the training set TS . An alternative definition of the semantic loss function we used in the second version of the RS2D system is presented in [4] together with the results of its experimental evaluation. As expected, both RS2D versions significantly differ in the quality of their results. In particular, the second version improved the retrieval performance of the original one in terms of precision but at the cost of increased communication efforts.

2.4 Learning of Individual Q/A Behavior

Each peer learns the most probable query answering (Q/A) behavior of each of its direct neighbours individually in order to be able to decide on whether it is beneficial to forward queries to these peers, or not. In particular, the conditional probability $P(c_j|x)$ of possible answering result classes of the considered peer v_k based on its observed Q/A behavior in the past is computed based on the prior probability $P(x|c_j)$, the likelihood $P(c_j)$, and the normalizing evidence factor $P(x)$ from the training set TS , with

$$P(c_j) = \frac{|TS_{c_j}|}{|TS|}; P(x|c_j) = \prod_{l=1}^n P(x_l|c_j); P(x) = \sum_{j=1}^{|C|} P(x|c_j) \cdot P(c_j) \quad (4)$$

and the feature probability $P(x_l|c_j) = \frac{|TS_{x_l,c_j}|}{|TS_{c_j}|}$ of the occurrence of the observation feature component x_l for given class c_j .

The decision making process heavily relies on the training set TS that each peer maintains individually. Initially, when a peer joins the network, its training set TS is empty; in this case, it sends its queries to all its direct neighbours until the size $(\theta(TS))$ of its training set, more specifically $TS_{fid,tid}$ is sufficiently large for continuing with risk assessment driven routing decisions from this point. Our experiments provide evidence in favor of $\theta(TS_{fid,tid}) = 8$.

2.5 Mixed Risk Based Query Routing Decision

The risk assessment driven routing decision of each peer v whether to route a given query q to a given neighbour v_k , or not, is then based on the mixed conditional Bayesian risk of doing so in terms of both semantic gain and communication costs. It is computed as follows

$$R(\alpha_i|x) = \sum_{j=1}^{|C|} \lambda(\alpha_i, c_j) \cdot P(c_j|x) \quad (5)$$

It sends the request $r = (q, S_q, S'_q, a)$ based on its actual training set TS to v_k , if the risk of forwarding r to v_k is minimal. Otherwise, it rejects a received request r , if it has been already processed locally, or a fixed number of forwarding steps (hops) is reached, or the risk of further forwarding is maximal for each of its neighbours. As a consequence of routing r only to those peers for which the corresponding alternative with minimal risk

$$\alpha^* = \operatorname{argmin}\{R(\alpha_0|x), R(\alpha_1|x)\} \quad (6)$$

is α_1 , the overall risk $R = \int R(\alpha(x)|x)P(x)dx$ is also minimal.

For each request r , each peer collects the replies it receives from its neighbours for r , that is the set of top- k semantically matching services, and merges them together with its local answer set. The top- k services of the merged set with semantic gain is then recursively returned on the unique return path to the one who did forward the request. That is, the complete answer set for a query is created while being propagated back to its origin. At the same time, each peer involved in this process continuously learns about the query answering behaviour of each of its neighbours in general. It caches the individual observations in its local training set each time it receives a reply. This, in turn, enables each peer to estimate the corresponding risk of forwarding a query to individual peers.

The computational complexity of RS2D is $O(nm \cdot T_\sigma + nm \cdot \log k + nt)$ with n and m of peers, respectively, services in the network, and t and k maximal size of the training, respectively, answer sets. That is, the adaptive RS2D search is linear with respect to the routing decision but subject to the computational complexity $O(T_\sigma)$ of the used matchmaker in total; the proof is given in [2].

2.6 Join/Leave Operations

RS2D requires minimal message exchange on dynamic operations such as node arrivals or departures. The join-operation of a peer in RS2D enabled P2P networks is implemented as a simple handshake-advertisement: Each peer that wants to join the network, broadcasts a one-hop advertisement ($\text{TTL} = 1$) to all peers in its neighbourhood, and then waits for acknowledgement-messages. If at least one peer answers, the requesting peer considers itself to be on line and part of the network, and both peers mutually take themselves into account for future routing decisions. The leave-operation is completely passive: A peer just drops out and stops answering to messages. Its neighbouring peers will detect its absence as soon as they attempt to send a new message to it, and consequently remove those training records from their local training sets that relate to it.

In its current version, the RS2D protocol does not explicitly enforce the propagation of changes in the expertise of each peer through the network. Such changes are caused by the insertion, deletion, or modification service descriptions. Rather, each RS2D peer gradually adapts to such changes each time it receives a response to a query from the owner of the changed service by

observing possibly implied changes of the answering behaviour including, in particular, the semantic gain.

3 Comparative Evaluation of RS2D

We have implemented the P2P service retrieval mechanism RS2D v1, and evaluated it by means of simulation. In the following, we present the results of the evaluation of the original RS2D protocol compared to the classic broadcast based routing (BCST), random selection of two neighbor peers (RND2), and Bibster like routing (BIBL) [10] based on peers that have prior knowledge on the semantic overlay network.

For testing purposes, we randomly generated unstructured, sparsely connected P2P networks of different size with 50, 100, 200, and 576 peers, and used the OWLS-TC2 service retrieval test collection [14] which contains 576 OWL-S services, 36 queries with relevance sets, and the OWLS-MX matchmaker [13] for semantic matching by each peer. In each simulation run, the queries are sequentially processed by each peer to generate the training set, and the top 20 services are returned by each peer only. The P2P service retrieval performance is measured in terms of micro-averaged precision and recall against communication overhead with respect to the maximum hop count for query propagation.

Testing of RS2D in large scale networks with thousands of peers requires, in particular, a correspondingly large and meaningful semantic web service retrieval test collection, which is not available yet. For details on the implementation of the RS2D simulator, as well as the improved RS2D v2, and the results of its experimental evaluation, we refer the interested reader to [4,2]. We were not able to run the GSD

3.1 Service Retrieval Performance

In our experiments, we evaluated two essential aspects of P2P service retrieval performance measurement:

1. Service distribution to peers: Uniformly at random Vs. Single peer hosts all relevant services per query
2. Query distribution to peers by the user: Random querying of peers Vs. One central Q/A peer, considered as exclusive querying interface to the system for the user

For reasons of space limitation, we present only the representative experimental results, and refer the interested reader for complete set of our RS2D performance and robustness experiments to the RS2D project web page [3].

Experiment 1: As figure 2 shows, in a network of 576 peers with evenly distributed 576 services, and random querying of peers, RS2D outperforms BIBL as well as RND2 in terms of precision with lesser number of hops which yields a

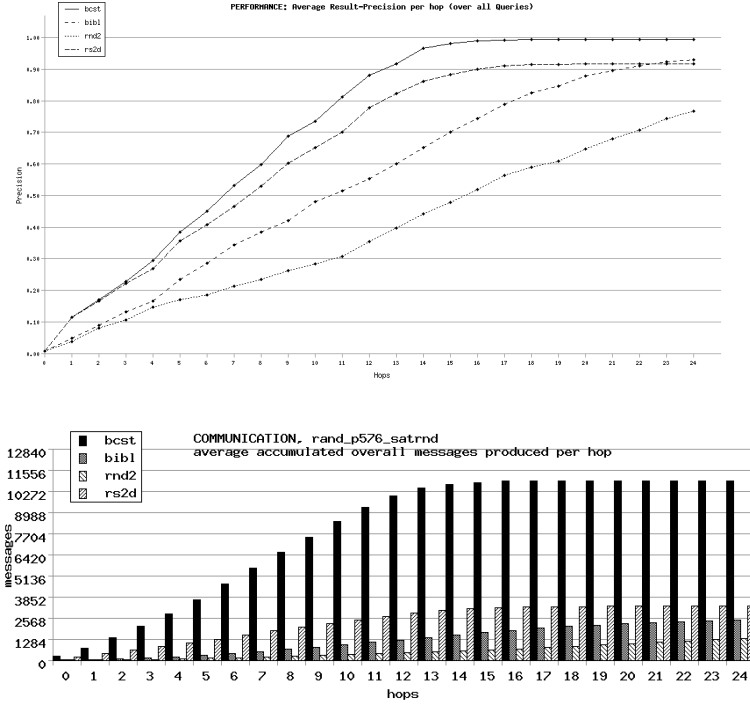


Fig. 2. Experiment 1, precision and communication

faster response time. However, after around 22 hops it gets slightly overtaken by BIBL due to its query-specific routing with prior knowledge on the peer expertises. The second version of RS2D ([4], fig. 2) performs totally different in that it could not be outperformed by BIBL at all while being almost close to optimal from the beginning. Each of both results were confirmed by the respective version of RS2D also for smaller P2P networks (50, 100, 200 peers).

When it comes to communication overhead, RS2D performs as bad as broadcast based routing in the initial training phase. In fact, RS2D does a multicast in the initial training phase, though this phase may be very short (eight recorded replies in average were sufficient to initialize successful risk driven query routing), so BIBL outruns RS2D because it exploits its prior semantic overlay knowledge for routing. However, the situation changes when processing the last query of the test collection: RS2D is faster than the initially more savvy BIBL (see fig.3). The same holds for RS2D v2 ([4], fig. 4), though RS2D v1 turned out to be much more efficient in this respect, hence faster in its response time.

Experiment 2: We also simulated the case of single query authorities, i.e., one peer hosts all the relevant services to a query, with random querying of the peers.

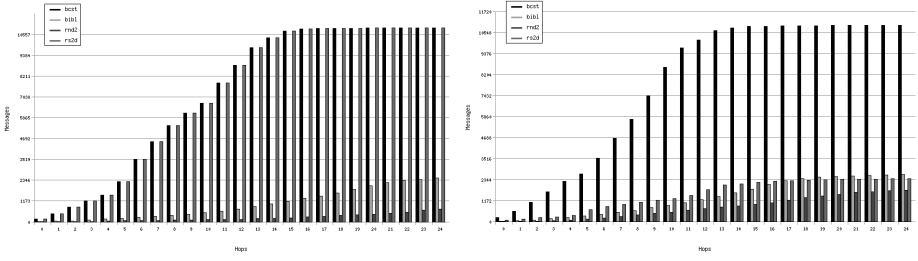


Fig. 3. Experiment 1, communication, first/last query

For each query a different but distinguished peer was chosen at random to host the relevance set. The results (fig. 4) show that BIBL eventually outperforms RS2D, because it can exploit its knowledge on the semantic overlay for optimal query-specific routing. RS2D v1 gets outperformed even by RND2 in the end. Main reason for that is that relying only on the *average* query answer behaviour renders it very difficult to find the single authority for each individual query. In any case, this clear deficiency of RS2D v1 motivated us to let each peer adapt according to a differently designed semantic loss function in the second version of RS2D ([4], fig. 5) which then performed almost optimal from the beginning, strongly outperforming both BIBL and RND2, but at the cost of (reasonably) higher communication efforts.

Experiment 3: We also tested RS2D in a setting with one distinguished (central querying) peer executing all queries on behalf of a user with 576 services distributed uniformly at random in a 576 peer network, and initial size of training set $\theta_{TS} = 8$.

According to the experimental results (cf. fig. 5), RS2D performs optimal (curve is on that of BCST) in terms of precision but drastically reduced communication overhead. The same was experimentally shown to hold for the performance of RS2D v2 ([4], fig. 7) with comparably minor reduction of communication only. Main reason for the good performance in this case is that the central RS2D peer is able to directly search the minimal spanning tree for all queries after its initial multicast.

In fact, further evaluation revealed that for the same reason the observed deficiency of RS2D in case of authorities in an unstructured network with uniform at random query and service distribution (cf. fig. 4) can be largely mitigated by the introduction of a central query-answering peer (cf. fig. 6). In this case, RS2D remains to perform optimal whereas both savvy BIBL and random walk RND2 are able to eventually close up, though BIBL gets partially outperformed by RND2. Interestingly, RS2D gets even faster than its competitors in terms of communication cost. This setting also turned out to be the best for RS2D in terms of achieved performance and communication according to our simulations.

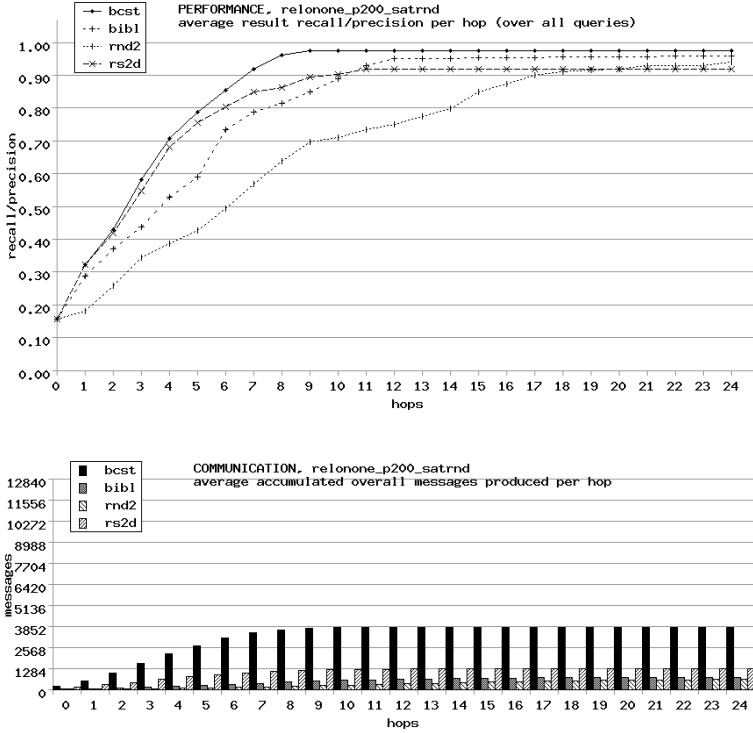


Fig. 4. Experiment 2, precision and communication (query authorities, $n = 200$)

3.2 Robustness

We tested the robustness of RS2D against dynamic changes of the topology of a network of 576 peers.

Experiment 4: During simulation peers may randomly join or leave the network with a rate of about one such operation for each five simulation steps, with around 80% of all peers ($= 460$) being on line in average.

If a RS2D peer is not able to route a return message to relevant peers according to the return path because they are not reachable for some reason, it tries to contact the subsequent peers in the path to establish a detour for this case. If this strategy fails, it issues a limited 2-hop broadcast to all of its neighbours. Only if the peer turns out to be isolated, or none of the contacted peers is able to forward the return message to any of the subsequent ones in the respective return path, the return message is discarded yielding a total loss of all related intermediate results of the query. The experimental results show that albeit the implied communication effort of RS2D is higher than that of BIBL compared to the static cases, it remains to be more efficient in terms of performance except the optimal BCST. Not surprisingly, the performance of all systems went down

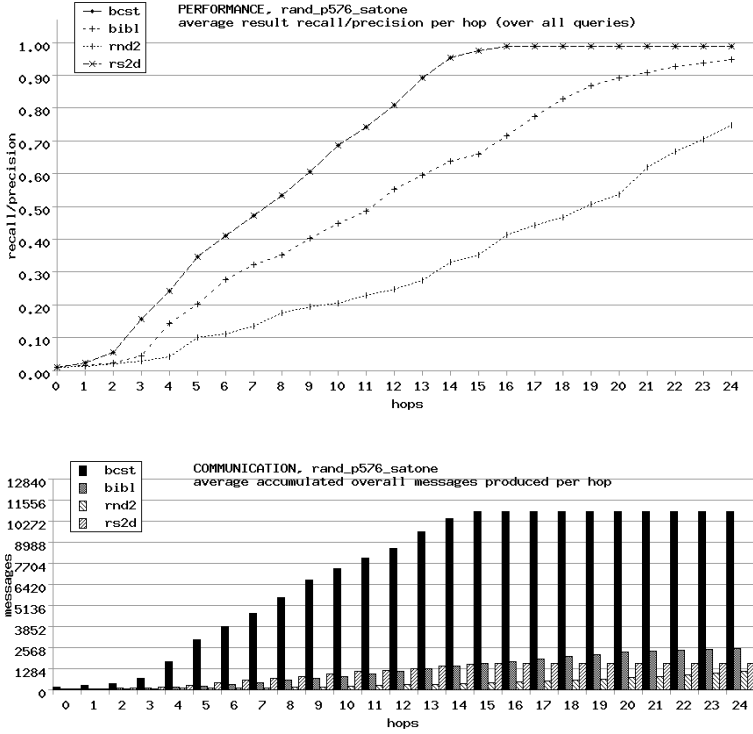


Fig. 5. Experiment 3a, precision and communication (central Q/A peer, random)

compared to those in the static cases; the second version of RS2D performed slightly better than the original one also in the dynamic case.

4 Related work

Relevant work includes approaches to probabilistic adaptive object retrieval in unstructured P2P networks such as in [15,11,8] as well as systems that do exploit semantic web technologies for this task such as Bibster[10] and GSD[5].

In APS [15], each peer forwards a query to k of its N neighbors. If a hit (match) occurs, the request terminates successfully, otherwise it gets forwarded to only one of this peers neighbors. This procedure continues until all k requests (with the same query) have terminated, either with success, or failure. Each APS peer keeps a local index with one value for each object (service) it requested, or forwarded a request for, per neighbor. Along the paths of all k queries, indices are updated as follows. In the pessimistic learning approach, each peer on a querys path decreases the respective object index value by a fixed amount reflecting the relative probability of its being the next hop choice in a subsequent search for the same object concurrently with the actual search. If the query succeeds by

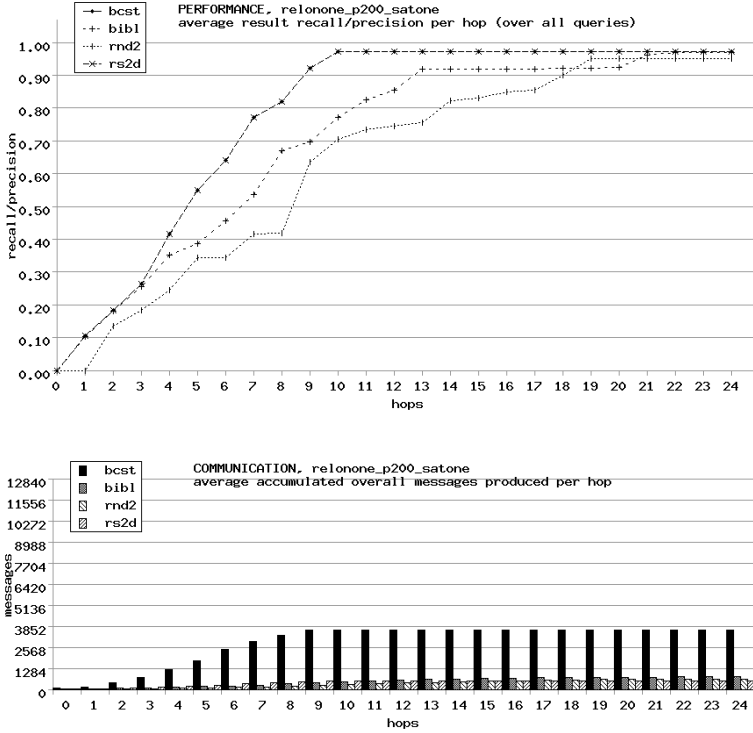


Fig. 6. Experiment 3b, precision and communication (central Q/A peer, authorities, $n = 200$)

eventually locating (an exact copy of) the object at some peer, the respective index values of all peers in the return path get updated by a fixed amount larger than the previously subtracted amount (positive feedback). The optimistic search operates in an opposite fashion (negative feedback) after a query fails. Hence, the initial probability of a peer for a certain object increases if the object was discovered through (or at) that node, and decreases otherwise. In contrast to RS2D, APS peers perform an informed exact object search only with index update by same amount of values for each peer. Hence, APS does not exploit different qualities of semantic matching at individual peers, nor the combined use of semantic gain and estimated communication efforts to drive the search like in RS2D. Finally, RS2D's minimal risk driven routing scheme is based on the average but not object specific Q/A behavior of peers.

The GSD routing scheme [5], in contrast to RS2D, relies on an acyclic ontology to be used by each peer to categorize its services by associating their identifiers with appropriate concepts. The degree of semantic relevance of any service to a given query is assumed to be inverse proportional to the computed distance between their concepts in the ontology.

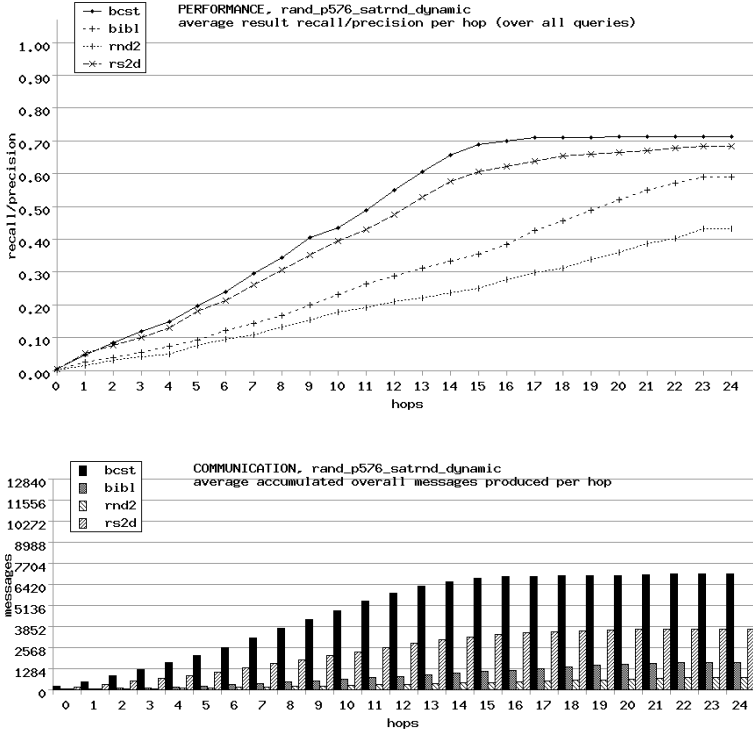


Fig. 7. Experiment 4, precision and communication

Bibster’s object specific search [10], in contrast to RS2D, exploits prior knowledge about the expertise of peers. In an initial advertisement round, each peer caches the advertised services of its neighbours only if they are semantically close to at least one of its own. Any query is then forwarded only to two neighbours with maximum matching expertise. The additional communication overhead induced by the initial expertise building is not included in the comparative evaluation results shown in section 4.

5 Conclusion

This paper presents RS2D v1, an approach to adaptive and mixed probabilistic risk driven search for OWL-S services in unstructured P2P networks in the semantic web. Experimental results showed that RS2D is fast and robust with comparatively high precision. It is, however, weak in finding single query authority peers, and its scalability to large scale networks with more than 576 service peers remains to be investigated due to lack of a sufficiently large OWL-S service retrieval test collection. In [4], we showed that the performance of RS2D v1 can be significantly improved by means of a different semantic loss function but at

the cost of communication efforts. RS2D will be made available to the semantic web community under LGPL-like license at semwebcentral.org.

References

1. S. Androutsellis-Theotokis and D. Spinellis. *A survey of peer-to-peer content distribution technologies*. ACM Computing Surveys, 36(4):335371, 2004
2. U. Basters. *RS2D - Risikobasierte semantische Suche nach OWL-S Diensten in unstrukturierten P2P Netzen*. Diploma thesis, University of the Saarland, Computer Science Department (in German).
3. U. Basters. RS2D v1.0 and RS2D v2.0 online: <http://www.basters.de/rs2d/>.
4. U. Basters and M. Klusch. *Risk Driven Semantic P2P Service Retrieval*. Proc. 6th Int. IEEE Conference on P2P Computing, Cambridge, UK, 2006.
5. D. Chakraborty, A. Joshi, T. Finin, and Y. Yesha. *GSD: A novel groupbased service discovery protocol for MANETS*. Proc. 4th IEEE Conference on Mobile and Wireless Communications Networks (MWCN), 2002.
6. Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker. *Making Gnutella-like p2p systems scalable*. Proc. Int. Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, Karlsruhe, Germany, 2003.
7. C. Elkan. *Boosting and Naive Bayesian learning*. Technical report, Department of Computer Science and Engineering, University of California, San Diego, 1997.
8. R. Ferreira, M. Ramanathan, A. Awan, A. Grama, S. Jagannathan. *Search with Probabilistic Guarantees in Unstructured Peer-to-Peer Networks*. Proc. Int. IEEE Conference on P2P Computing, Konstanz, Germany, 2005.
9. Gnutella. <http://gnutella.wego.com/>.
10. P. Haase, R. Siebes, and F. van Harmelen. *Expertise-based Peer selection in Peer-to-Peer Networks*. Knowledge and Information Systems, Springer, 2006.
11. V. Kalogeraki, D. Gunopulos, and D. Zeinalipour-Yazti. *A local search mechanism for peer-to-peer networks*. Proc. 11th Int. Conference on Information and Knowledge management (CIKM), ACM Press, 2002.
12. M. Klusch, B. Fries, and K. Sycara. *Automated Semantic Web Service Discovery with OWLS-MX*. Proc. 5th Intl. Conference on Autonomous Agents and Multiagent Systems (AAMAS), Hakodate, Japan, 2006
13. OWLS-MX: <http://projects.semwebcentral.org/projects/owls-mx/>.
14. OWLS-TC: <http://projects.semwebcentral.org/projects/owls-tc/>.
15. D. Tsoumakos and N. Roussopoulos. *Adaptive Probabilistic Search (APS) for Peer-to-Peer Networks*. Proc. Int. IEEE Conference on P2P Computing, 2003.