Risk Driven Semantic P2P Service Retrieval

Matthias Klusch, Ulrich Basters
German Research Center for Artificial Intelligence (DFKI)
Saarbruecken, Germany
klusch@dfki.de, ulrich@basters.de

Abstract

In this paper, we present a novel approach, named RS2D, to risk driven semantic service query routing in unstructured, so called pure P2P networks. Following the RS2D protocol, each peer dynamically learns about the query answering behavior of its direct neighbours. without prior knowledge on the semantic overlay. The decision to whom to forward a given service request is then driven by the estimated mixed individual Bayes' conditional risk of routing failure in terms of both semantic loss and high communication costs. The results of our experimental evaluation of retrieval performance and robustness show that RS2D top performs compared to other relevant systems.

1. Introduction

The retrieval of relevant services is one key to service oriented computing in the web and semantic web. As of today, web services are supposed to be still discovered mainly by means of central repositories or registries such as UDDI [1]. In contrast, unstructured P2P service networks are expected to be robust against dynamic changes in the underlying network topology at the very expense of administrative communication overhead in due course of the self-regulation of the peers. The major challenge of decentralized semantic Web service retrieval in unstructured P2P service networks is to keep the communication costs of service retrieval low with reasonably high precision of the returned results.

Different approaches to solve this problem have been proposed in the literature; an accessible survey is provided in [2]. Whereas broadcast-based approaches are very robust with high precision they typically suffer from poor scalability due to their high communication overhead. Randomized routing usually keeps the communication effort low, but at the expense of low precision of the returned results. Our solution to this problem is the first risk assessment driven semantic service query routing protocol, named RS2D. Key idea is to let the peers dynamically learn the average query-answer behavior of their direct neighbours in the network for making individual probabilistic risk based routing decisions with respect to both semantic gain and communication costs. In contrast to other existing approaches, RS2D does not require any prior knowledge on the environment including service distribution, global ontologies, or network topology. We implemented the RS2D protocol and experimentally evaluated its performance and robustness in randomly generated unstructured P2P networks in different scenarios.

The remainder of this paper is organized as follows: After brief discussion of related work in section 2, we provide the outline of the RS2D protocol in section 3, and provide the details of the underlying Bayesian risk based routing decision rule in section 4. Section 5 provides and discusses the results of the experimental evaluation of RS2D compared with other relevant approaches. Section 6 presents some insights into the implementation of the RS2D system and its simulation, and section 7 concludes the paper.

2. Related Work

The GSD-algorithm by Chakraborty et al.[4] takes advantage of the hierarchical structure of a global underlying ontology of the semantic network. Peers advertise their services not as service description, but with an ontological classification. When a peer gets a request for a service, it uses this classification to determine the distance between the requested and the offered services in the ontology tree. This approach has the clear disadvantage that only a static and commonly known ontology can be used. Additionally, sometimes the ontological classification might not be sufficient to find matching services, e.g. if they differ in their input and output parameters.

Another approach was proposed by Haase et al. with Bibster [8]. Their system relies on service advertisements that build up a semantic topology overlay. This is done by a special advertisement caching policy: peers add advertised services of neighbours to their list of known services only if they are semantically close to at least one of their own services. This way, peers become experts for semantically similar services. When a query then asks for a certain service, Bibsters routing mechanism chooses those two neighbours whose expertise is closest to the query. Thus the query travels along a path of peers with similar expertise what increases the result precision and decreases communication overhead. However, the message traffic induced by the initial exchange of service advertisements is rather high. Also, prior knowledge about other peers' ontologies as well as their mapping to local ontologies is assumed.

To the best of our knowledge, there exist no other relevant and implemented solutions to the problem of decentralized semantic service retrieval in unstructured P2P networks.

3 RS2D Routing Protocol Overview

One major challenge of decentralized service retrieval in unstructured P2P networks, is to achieve a reasonably high retrieval performance with low communication costs without any prior knowledge about the environment including services, ontologies, or network topology. There is no central directory or repository in the system. The basic idea of our solution to the problem is to allow each peer to quickly learn which of its direct neighbours in the network will probably return relevant semantic web services for a given query with minimal risk of both semantic loss and high communication in total. We first outline the RS2D protocol to be followed by each peer, and then provide the details of it in subsequent sections.

Let be for each peer v, q a service request (query); S set of locally known services, S_q the current top-k relevant services (URIs) retrieved; $a \in R$ the communication effort of propagating q, that is the number of messages in the routing subtree for q in the network graph; TS the individual training set of a peer consisting of information about previous queries and their results; $hop \in \mathbb{N}$ the distance from v in the network. Then, each peer v performs the following steps:

- Determine the set S'_q of services that are semantically relevant to q: $S'_q = S_q \cup \{\forall s \in S : \sigma(s,q)\}$.

  The function $\sigma(s,q) \in [0,1]$ maps the matching results of the used semantic web service matchmaker to $[0,1]$, where $\sigma(s,q) = 0$ and $\sigma(s,q) = 1$ represent a matching failure and exact match, respectively. For our experiments with RS2D, we used the hybrid OWL-S service matchmaker OWLS-MX [9] which renders RS2D independent from any fixed global ontology, as this matchmaker dynamically maintains a local matchmaker ontology by means of logic based reasoning upon provided service advertisements and requests (see also sect. 6).

- For each peer v_k in the direct neighbourhood of v ($hop = 1$):
  - Estimate the expected semantic gain $E(y)$, and communication costs $E(a)$ of forwarding request $r = (q, S_q, S'_q, a)$ to v_k based on the actual training set TS.
  - Compute the individual Bayes' conditional risk of routing r to v_k, or not (cf. section 4).
  - Send r to v_k, if the risk of forwarding is minimal, or if (initially) $TS = \emptyset$ then multicast r to all neighbours.

- Observe the query answer behavior of neighbour peers v_k by storing received replies with a semantic score $L(S''_q)$ of intermediate results S''_q returned and communication costs a per query in the local training set TS. The semantic score measures the quality of the set of retrieved services with respect to q by means of $L(S_q) := \sum_{s \in S_q} \sigma(s,q)$.

- Reject 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.

- Return set of top-k semantically matching services in a priority queue if the semantic gain is positive, that is $L(S''_q) - L(S'_q) > 0$.

Each peer collects the replies on query q it receives from its neighbours and merges them together with its local results set which is then returned to the one who did forward q to it. This way, the result 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.

4 Bayes Risk of Query Routing

The decision of each peer to route a given query q to any of its neighbours v_k is based on the individual estimated mixed risk of doing so in terms of both semantic gain and communication costs. The estimated semantic gain $E(y)$, the estimated communication costs $E(a)$ as well as the probability with which a neighbour will answer are computed from the training set TS by means of a naive Bayes approach [5]. More concrete, the risk assessment driven routing decision bases on the computation of the individual Bayes' conditional risk defined as:

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

with

- Binary routing alternatives α_0 and α_1 for not routing, respectively, routing the query.
- Query answer class set $C = \{c_0, c_1\}$ with classes c_0 (= query rejected because it already was processed by v_k) and c_1 (= v_k answers to the query with a semantic gain, i.e. with $L(S''_q) - L(S'_q) > 0$.
- Observation x of query answering behavior of v_k for past queries
- Mixed semantic and communication loss $\lambda(\alpha_i, c_j)$ for routing alternative α_i and query answer class c_j.
- Conditional probability P of query answering class c_j for given observation x.

Having computed the mixed risk values for each binary routing alternative for each of its neighbours, the peer then routes the query q only to those peers for which the corresponding alternative with minimal risk

$$\alpha^* = argmin\{R(\alpha_0|x), R(\alpha_1|x)\} \quad (2)$$

is α_1, otherwise rejects. This minimizes the overall risk $R = \int R(\alpha(x)|x)P(x)dx$ in compliance with the known Bayes Decision Rule, in other words a decision for the alternative with minimal overall risk is optimal.

What does an individual peer observe in concrete terms? From each reply to a given query q it received from some 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 a 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

fid: Identifier of the peer from which the request was received.

tid: Identifier of the peer to which the request was forwarded.

c_j: Query answer result class (c_0 or c_1).

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 in different ways. Let $TS_{p_1,\ldots,p_z} \subset TS$ denote the set of training records t with parameters p_1 to p_z set to given values, for example, $TS_{fid,tid}$ the subset which has the given values for fid and tid (here: $z = 2$ having $p_1 = fid$ and $p_2 = tid$).

The estimated semantic loss of routing 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 computed as follows:

$$\begin{array}{c|c|c} & \lambda(\alpha_0|\cdot) & \lambda(\alpha_1|\cdot) \\ \hline c_0 & -E(a)\kappa & 2\kappa \\ c_1 & \tau E(y) & -\tau E(y) \end{array} \quad (3)$$

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:

$$E(y) := \frac{1}{|TS_{fid,tid}|} \sum_{t \in TS_{fid,tid}} [L(S''_q)]_t - [L(S'_q)]_t \quad (4)$$

$$E(a) := \frac{1}{|TS_{fid,tid}|} \sum_{t \in TS_{fid,tid}} [a]_t \quad (5)$$

with $[x]_t$ extracting the parameter x from observation record t in the training set TS. The real-valued user preference parameter τ denotes the weighted relation between maximum semantic gain ($y = 1$) and communication costs the user is willing to accept; in our experiments, we obtained the best results with $\tau = 1000$. Each of the above defined cases of semantic loss of a routing decision by an individual peer v with respect to forwarding a given request to one of its neighbor peers v_k is justified as follows:

$\lambda(\alpha_0|c_0)$ No routing of the request to the targeted peer v_k takes place, but it would have been rejected by this peer anyway. As a consequence, the risk based decision is of benefit for v in terms of saved communication efforts ($-E(a)\kappa$).

$\lambda(\alpha_0|c_1)$ In this case, peer v does not forward the query to v_k but would have received a positive reply with semantic gain. Hence, the loss is computed in terms of the costs of the lost opportunity, that is the semantic gain weighted by its relation to individually prefered upper bound of communication costs ($\tau E(y)$).

$\lambda(\alpha_1|c_0)$ Peer v decides to route the query to v_k which rejects it. Hence, the decision was not beneficial for v in that it produced unnecessary communication costs of the specific request and reply.

$\lambda(\alpha_1|c_1)$ The request of peer v will be answered by v_k with some expected semantic gain for v. Hence, the decision is beneficial for v in terms of negative loss (utility $-\tau E(y)$).

Using λ for computing the risk of routing alternatives α_0, α_1 does reflect the classical loss relation between utility and costs:

$$R(\alpha_0|x) = -E(a) \cdot \kappa \cdot P(c_0|x) + \tau \cdot E(y) \cdot P(c_1|x) \quad (6)$$

$$R(\alpha_1|x) = 2 \cdot \kappa \cdot P(c_0|x) - \tau \cdot E(y) \cdot P(c_1|x) \quad (7)$$

Alternatively, one could have defined the semantic loss of the query answering class c_1 directly as difference between expected semantic gain and average communication costs in terms of number and volumes of messages exchanged:

	$\lambda'(\alpha_0\|\cdot)$	$\lambda'(\alpha_1\|\cdot)$
c_0	$-E(a)\kappa$	2κ
c_1	$E(y) - E(a)\kappa$	$-E(y) + E(a)\kappa$

(8)

However, according to the results of our experimental evaluation, this option can be significantly improved by the one chosen in terms of retrieval performance with only slightly increased communication efforts.

Then, 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 as usual based on he prior probability $P(x|c_j)$, the likelihood $P(c_j)$, and the normalizing evidence factor $P(x)$ from the training set TS:

$$P(c_j|x) = \frac{P(x|c_j) \cdot P(c_j)}{P(x)} \quad (9)$$

with

$$cP(c_j) = \frac{|TS_{c_j}|}{|TS|} \quad (10)$$

$$P(x|c_j) = \prod_{l=1}^{n} P(x_l, c_j) \quad (11)$$

$$P(x) = \sum_{j=1}^{|C|} P(x|c_j) \cdot P(c_j) \quad (12)$$

with the probability $P(x_l, c_j)$ of the occurence of the observation feature component x_l together with class c_j defined as

$$P(x_l, x_j) = \frac{|TS_{x_l,c_j}|}{|TS_{c_j}|} \quad (13)$$

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 $\theta(TS_{fid,tid}) = 1$ ($\theta_{TS} = 8$ when using the alternative semantic loss definition in equ.(8)).

5 Evaluation

We have implemented the RS2D protocol and evaluated it by means of simulation. For this purpose, 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 [6] which contains 576 OWL-S services, 36 queries with relevance sets, and the OWLS-MX matchmaker [7] for semantic matching by each peer.

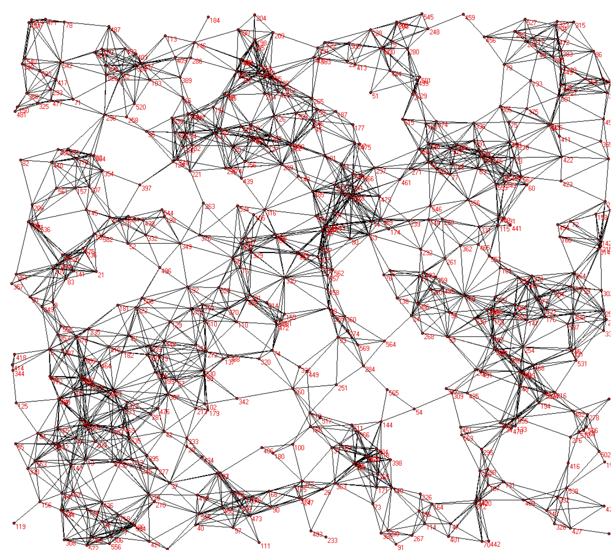

Figure 1. Example of unstructured network of 576 peers used in our experiments

In each simulation run, the queries are sequentially processed by each peer to generate the training set, and the top k (k = 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.

We evaluated the performance of the RS2D service query routing mechanism against the following relevant alternative approaches:

BCST Classic broadcast based routing forwards the query to all direct neighbours until a maximal number of hops is reached, or all neighbours reject the query. BCST always yields optimal precision but at the very expense of communication efforts.

RND2 Random peer selection: This method randomly selects two direct neighbours of a peer v_m to which the query is forwarded. RND2 has low communication costs but low precision as well. It is also used by developers of BIBL in [8] for the comparison of performance.

BIBL Bibster-like routing: This routing mechanism simulates the one used in the P2P system Bibster [8]. In particular, peers have prior knowledge about a fixed semantic overlay network that is initially built by means of a special advertisement caching policy. Each peer only stores those advertisements that are semantically close to at least one of their own services, and then selects for given queries only those two neighbours with top ranked expertise according to the semantic overlay it knows in prior.

5.1 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, as a single entry point to the system for the user

For reasons of space limitation, we present only representative results of selected experiments.

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 better response time. The same results can be obtained for RS2D in smaller networks.

However, unlike BCST this nearly optimal performance of RS2D does not come at the very expense of communication but only almost one third and twice of that of BCST and BIBL, respectively, as shown in figure 3.

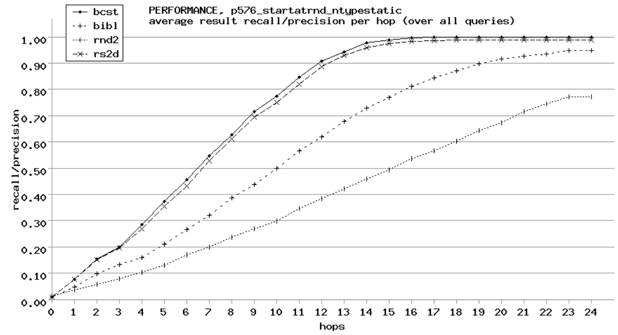

Figure 2. Experiment 1, Precision: RS2D outperforms BIBL and RND2, and performs close to optimal BCST.

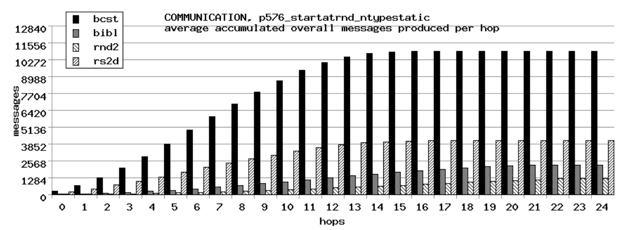

Figure 3. Experiment 1, Communication.

In more detail, RS2D performs as bad as broadcast in its initial training phase while in case of processing the last query of the test collection in one simulation run, RS2D outperforms even the more savvy BIBL system (see fig.4).

Please note that this provides evidence in favor of mixed risk-driven routing based on learned average Q/A behavior rather than query-specific routing only. It would be interesting to investigate a mix of both approaches in future.

Experiment 2: We also simulated the case of single query authorities and random querying of peers. In this case, one peer hosts all services that are relevant to a specific query, thus possesses the complete relevance set of this particular query. For each query a different peer was chosen at random. Then the queries were executed uniformly at random from different peers as in the first experiment.

In this case, BIBL is more efficient than its competitors as it heavily benefits from the exploitation of the given semantic overlay structure for optimal routing. RS2D is outperformed by BIBL because it is difficult to find the authority for a query when only the *average* query answer behaviour is considered (see fig.5).

Not surprising, in this case the communication costs of RS2D are higher than that of BIBL with given semantic

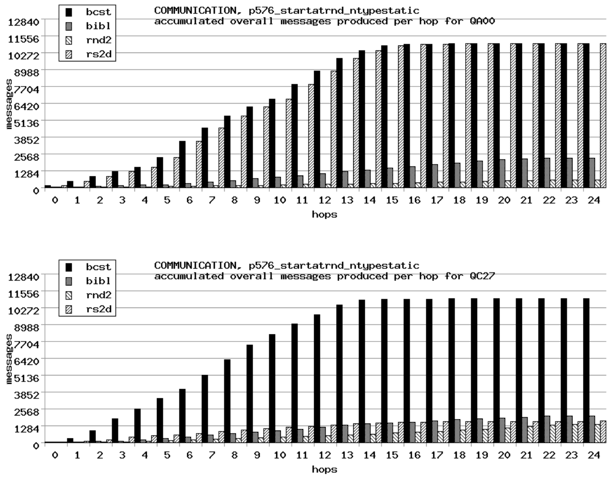

Figure 4. Experiment 1, Communication (first and last query): While in the initial training phase RS2D produces as much traffic as BCST does, it even outruns BIBL's traffic on the last executed query due to the learned average query answering behaviour. BIBL is more efficient in communication in the first run because of its exploitation of given semantic overlay knowledge for routing.

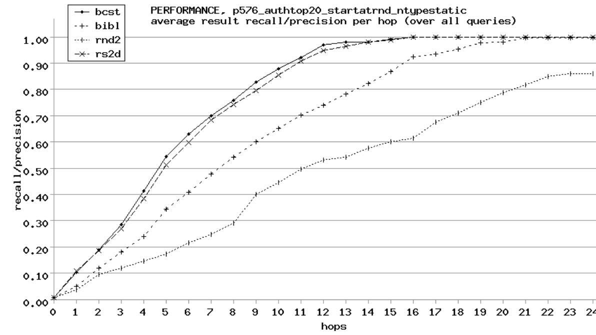

Figure 5. Experiment 2, Precision: RS2D still is almost optimal. BIBL exploits its given semantic topology to its maximum. Both outperform RND2.

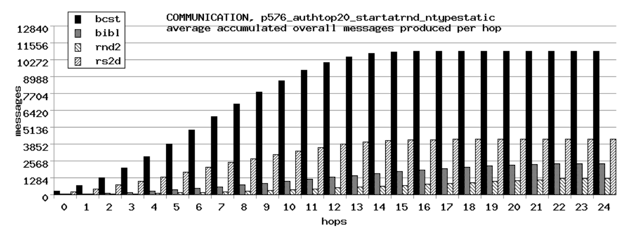

Figure 6. Experiment 2, Communication.

topology but still significantly lower than BCST as shown in figure 6.

Experiment 3: In case where one centrally located peer executes all 36 queries for the user, thereby acting as a single point of entry, with 576 services distributed uniformly at random in a 576 peer network, and $\theta_{TS} = 1$, RS2D performed as well as BCST in terms of precision (BCST-/RS2D curves are overlapping in fig.7) but drastically reduced communication overhead.

5.2 Robustness

The remaining question is how robust RS2D enabled unstructured P2P service networks are against dynamic changes, when peers can enter or leave the network at any time. For this purpose, we tested RS2D in a 576 peer network where each peer is hosting exactly one service but wiht only 80% of all the peers (= 460) being online. During the simulation run, we randomly let new peers join and leave the network with a rate of about one peer joining/leaving each 5 simulation steps (about 400 join/leave-operations per simulation run). In case of incomplete return paths for a query due to relevant peers having left the network, the peer in question tries to find the subsequent peers in the path. If this strategy fails, it sends out a limited 2-hop broadcast of the answer to its neighbours. If this last fallback also fails, the answer to the query is discarded yielding a total loss of all related intermediate results.

The join operation for a single 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 to all peers in its direct (one-hop) neighbourhood and then waits for acknowlegement-messages. If at least one peer answers, the requesting peer considers itself to be online, and both peers mutually take themselve for new routing decisions into account. The leave operation is completely passive: A peer just drops out and stops answering to messages. Each of its neighbouring peers will detect this as soon as it wants to send a new message to the dropped peer, and removes all training records that relate to this peer from the local training set.

As shown in figure 8, RS2D turned out to be reasonably robust against dynamic changes in the network topology. However, the precision went down for all tested routing mechanisms due to the following reasons. First, some of the relevant services are provided by offline peers, hence

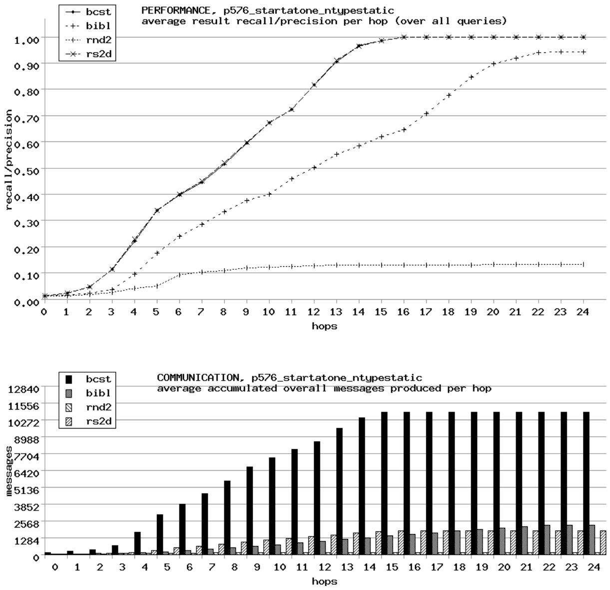

Figure 7. Experiment 3, precision: RS2D performs optimal (curve is on that of BCST). Central peer searches minimal spanning tree for all queries after initial multicast; Communication: RS2D outperforms BIBL with significantly lower communication effort.

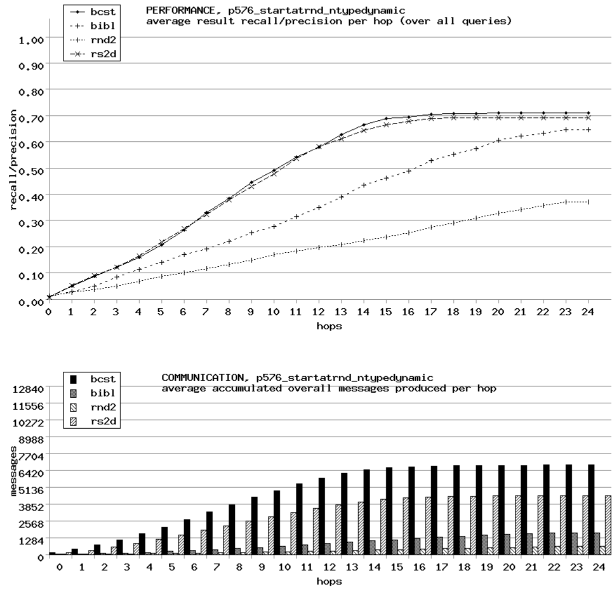

Figure 8. Experiment 4, dynamic: Precision goes down for all routing methods but RS2D significantly outruns both Bibster and random selection, hence is more robust to network-fluctuation. However, this comes at the very expense of communication overhead compared to Bibster, though half of that of Broadcast.

were unreachable. Second, some query answers were not propagated to the querying peer due to broken links in the network. Please note that the precision of BCST is optimal for this scenario, and RS2D is close to it but with only half of its communication efforts. This is because the the initial training phase is only repeated for recently joined peers - all stable peers are still risk-evaluated when taking the forward-decision.

Not surprising, Bibster-like routing performs poor in dynamic environments since its semantic topology breaks to the same extent the network topology changes. Building up a semantic topology is a very costly process as each peer has to advertise its hosted services at the cost of one advertisement message propagated over 3 hops in our experiments leading to a traffic load of about 212.000 messages in a 576 peer network, and about 5.800 messages for each of the 36 queries.

For more details on the RS2D performance and robustness experiments, we refer the interested reader to the RS2D experiment web page [3].

6 Implementation

The RS2D approach to OWL-S service retrieval in unstructured P2P networks has been implemented in Java 1.5, and evaluated by simulation on one server. The architecture of the simulator is shown in figure 9; the simulator also provides PHP-script based online visualization of the experimental evaluation results.

For computing the numeric semantic score $LS(S_q)$ used by RS2D for its risk based routing decision (cf. section 3), we defined a simple linear mapping ($\sigma(s,q)$) of the oputput of the semantic Web service matchmaker OWLS-MX [9] to the interval $[0,1]$ as shown in figure 10.

The OWLS-MX code is available at [7]. 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 according to five different filters and selected IR similarity metric. Logical subsumption failures produced by the integrated description logic reasoner of OWLS-MX are tolerated, if the computed syntactic similarity value is sufficient. What makes OWLS-MX particularly suitable to ser-

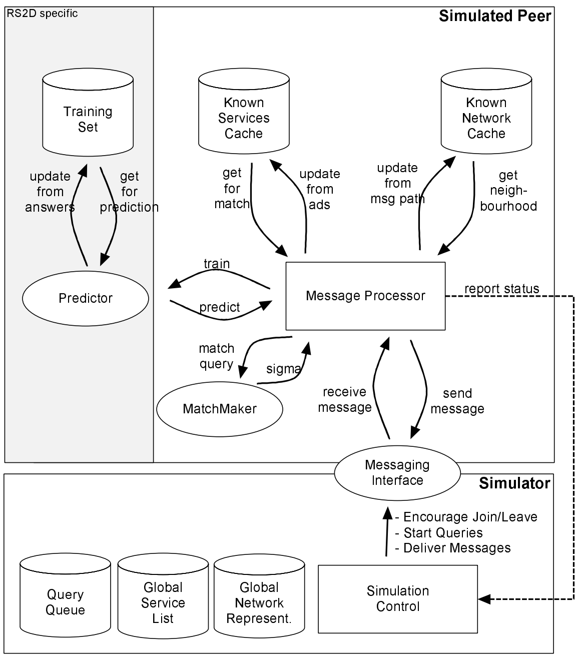

Figure 9. Architecture of RS2D Simulator.

vice retrieval in unstructured semantic P2P service networks is its capability to dynamically maintain a local ontology, hence renders RS2D independent from the use of any fixed global ontology like in GSD. It classifies arbitrary query concepts into its dynamically evolving ontology based on a commonly shared minimal basic vocabulary of primitive components instead of limiting query I/O concepts to terminologically equivalent service I/O concepts in a shared static ontology as, for example, the OWLS-UDDI matchmaker does.

For our experiments, we used the OWL-S service retrieval test collection OWLS-TC which is available at [6]. The collection consists of 576 OWL-S 1.1 services; its size in particular limited the maximum size of the unstructured P2P networks of our experiments as we neither did extend the collection nor distributed dummy service copies to peers for simulation.

7 Conclusion

We presented a first approach, named RS2D, to risk assessment driven semantic service retrieval in unstructured P2P networks without prior knowledge on the semantic overlay. It relies on the dynamic learning of averaged query-answer behavior of peers for minimal mixed routing risk. Experimental evaluation of RS2D showed that it is very robust and fast with reasonably high precision compared to

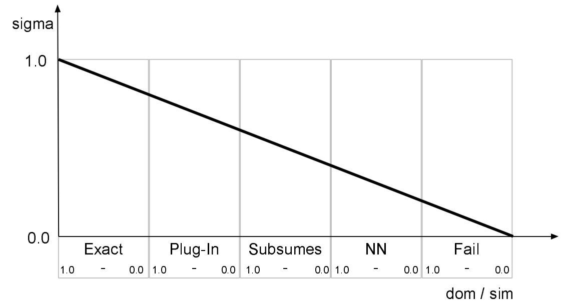

Figure 10. Used mapping of the degrees of semantic service matching returned by the OWLS-MX matchmaker to [0,1] for computing the numeric semantic score.

other existing relevant approaches. It is, however, weak in finding single query authority peers, and requires initial training, though only for an acceptable amount of time. We intend to make RS2D publicly available under GPL-like license at semwebcentral.org.

References

[1] *OASIS Standard consortium. Universal Description, Discovery and Integration (UDDI) protocol.* http://www.uddi.org/.

[2] S. Androutsellis-Theotokis and D. Spinellis. *A survey of peer-to-peer content distribution technologies.* ACM Computing Surveys, 36(4):335-371, December 2004.

[3] U. Basters. RS2D experimental evaluation results online: http://www.basters.org/rs2d/.

[4] D. Chakraborty, A. Joshi, T. Finin, and Y. Yesha. *GSD:A novel groupbased service discovery protocol for MANETS*. 4th IEEE Conference on Mobile and Wireless Communications Networks (MWCN), 2002.

[5] C. Elkan. *Naive Bayesian Learning, citeseer.ist.psu.edu/30545.html*.

[6] B. Fries and M. Klusch. OWL-S service retrieval test collection OWLS-TC v2, download at http://projects.semwebcentral.org/projects/owls-tc/.

[7] B. Fries and M. Klusch. OWLS-MX matchmaker source code, download at http://projects.semwebcentral.org/projects/owls-mx/.

[8] P. Haase, R. Siebes, and F. van Harmelen. *Expertise-based Peer selection in Peer-to-Peer Networks*. Knowledge and Information Systems, Springer Verlag, 2006.

[9] 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. ACM Press,, 2006.