

QUERY ROUTING IN A PEER-TO-PEER SEMANTIC LINK NETWORK

HAI ZHUGE

*China Knowledge Grid Research Group, Key Lab of Intelligent Information Processing,
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China*

JIE LIU, LIANG FENG, XIAOPING SUN, AND CHAO HE

*China Knowledge Grid Research Group, Key Lab of Intelligent Information Processing,
Institute of Computing Technology, Chinese Academy of Sciences;
Graduate School of the Chinese Academy of Sciences, Beijing, China*

A semantic link peer-to-peer (P2P) network specifies and manages semantic relationships between peers' data schemas and can be used as the semantic layer of a scalable Knowledge Grid. The proposed approach consists of an automatic semantic link discovery method, a tool for building and maintaining P2P semantic link networks (P2PSLNs), a semantic-based peer similarity measurement for efficient query routing, and the schema mapping algorithms for query reformulation and heterogeneous data integration. The proposed approach has three important aspects. First, it uses semantic links to enrich the relationships between peers' data schemas. Second, it considers not only nodes but also the XML structure in measuring the similarity between schemas to efficiently and accurately forward queries to relevant peers. Third, it copes with semantic and structural heterogeneity and data inconsistency so that peers can exchange and translate heterogeneous information within a uniform view.

Key words: Knowledge Grid, peer-to-peer computing, query routing, semantic link network.

1. INTRODUCTION

A P2P system includes a large, scalable, dynamic, autonomous and heterogeneous network, where nodes (peers) can exchange data and services in a completely decentralized and equal manner.

P2P networking supports knowledge management in a natural manner by closely adopting the conventions of face-to-face human communications. Knowledge exchange among peers is facilitated without the traditional dependence on servers (Tiwana 2003).

The original motivation for the early P2P systems is file sharing (Clarke et al. 2000). Search methods in P2P networks were surveyed in Wulff and Sakaryan (2004) and Tsoumakos and Roussopoulos (2003). Both the network structure and the search algorithm significantly influence the properties of P2P applications (Tsoumakos and Roussopoulos 2003).

Current P2P systems are of three kinds: (1) unstructured P2P systems, such as Gnutella (<http://www.gnutella.com>), where peers may join and leave the network without any notification; (2) structured P2P systems, where connections between peers are fixed and data placement is related to the structure formed by peer connections (Balakrishnan et al. 2003); and (3) hybrid P2P systems, such as Napster (<http://www.napster.com>), where file sharing is decentralized, but the file directory is centralized.

The unstructured P2P systems mostly employ flooding and random-walk search approaches to locate files. The flooding approach forwards a peer's queries to all its neighbors, which results in traffic problems. The random-walk approach forwards a peer's queries only to randomly selected neighbors. Each selected peer repeats this process until the required data are found. To be effective, a query-routing strategy should forward queries only to peers who are likely to match the queries.

The research work was supported by the National Basic Research and Development Program of China (973 project no. 2003CB317001) and the National Science Foundation of China (no. 60273020 and 70271007).

Address correspondence to Hai Zhuge, China Knowledge Grid Research Group, Key Lab of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100080, China; e-mail: zhuge@ict.ac.cn

To improve the performance of the flooding and random selection approaches, peer-clustering and peer-indexing approaches have been proposed. For example, peer-clustering approach was used to redistribute peers in super-peer networks to improve routing efficiency (Löser et al. 2003; Nejdil et al. 2003). The semantic small world overlay network for semantic-based P2P search was introduced in Li et al. (2004).

The objective of peer indexing is to allow peers to select from an index the “best” neighbor to send a query to, rather than flooding or random selection. For example, the content-based routing was proposed (Koloniari and Pitoura 2004), whereby information about the structure of documents is maintained while routing queries. Crespo and Garcia-Molina (2002) described three types of routing index based on document classification: compound, hop-count, and exponential. Nakauchi, Morikawa, and Aoyama (2004) applied a general approach to semantic keyword search based on keyword relationship and query expansion.

Structured P2P systems, such as Can (Ratnasamy et al. 2001), Chord (Stoica et al. 2001), Pastry (Rowstron and Druschel 2001), and Tapestry (<http://www.cs.berkeley.edu/~ravenben/tapestry/>), use a distributed hash table (DHT) for routing.

DHT-based P2P systems are not suitable for complex queries because they only support keyword-based and exact-match lookup. Issues related to complex queries in DHT-based P2P systems were outlined in Harren et al. (2002). Some open questions about DHT-based routing algorithms were raised (Ratnasamy, Shenker, and Stoica 2002). The content- and semantic-based P2P information retrieval system, *pSearch*, was introduced (Tand, Xu, and Mahalingam 2003). It combines the efficiency of DHT systems and the accuracy of the state-of-the-art IR algorithms. Mechanisms for locating data using incomplete information in P2P DHT networks were introduced (Garces-Erice et al. 2004), where multiple and hierarchical indices were used to help users to locate data even from poor information in the query.

Previous research on peer data management systems mainly considers data models for P2P databases, peer schema mediation methods, and query reformulation and optimization algorithms. Results mainly include the P2P-based system for distributed data sharing and management (Ng et al. 2003; Ooi, Shu, and Tan 2003), the local relational model for mediating between peers in a PDMS (Bernstein et al. 2002), the architecture supporting data coordination between peer databases (Giunchiglia and Zaihrayeu 2002), the approaches to automatic schema matching (Rahm and Bernstein 2001), the algorithm for mapping data in P2P systems (Kementsietsidis, Arenas, and Miller 2003), the solution to achieving semantic agreement in the P2P network (Aberer, Cudre-Mauroux, and Hauswirth 2003), the generic schema-matching prototype Cupid (Madhavan, Bernstein, and Rahm 2001), the query reformulation algorithms for XML-based peers (Deutsch and Tannen 2003; Halevy et al. 2003a; Halevy et al. 2003b), the approach to optimizing query reformulation in a PDMS (Tatarinov and Halevy 2004), and the ontology-based P2P system for exchanging bibliographic data among computer science researchers (Broekstra et al. 2004).

The main purpose of the Knowledge Grid is to share and manage globally distributed knowledge resources in an efficient and effective way (Zhuge 2004). P2P networks can be adopted as the semantic overlay layer of the Knowledge Grid. The scalability and autonomy make the P2P network a promising underlying infrastructure for a scalable Knowledge Grid. Integrating and managing heterogeneous knowledge and information in large-scale P2P networks poses the following three key problems:

1. Identifying semantically relevant peers autonomously.
2. Routing a query accurately and efficiently from the originating peer to relevant other peers to reduce network flooding.

3. Integrating heterogeneous knowledge returned from different peers to provide users and other peers with a uniform data usage mode (P2P systems do not have a global schema like traditional data integration systems).

Although there have been some projects using semantic-based routing in P2P networks (Crespo and Garcia-Molina 2002; Löser et al. 2003; Nejdil et al. 2003; Koloniari and Pitoura 2004), neither do they semantically identify relationships between peers, nor do they work well in the above three key problem areas.

This paper proposes an unstructured P2P semantic link network (P2PSLN) to solve the first problem. The semantic relationships between peers' data schemas are specified through semantic links as defined by Zhuge (2003). Peers are encapsulated as soft devices, the software service mechanisms introduced by Zhuge (2002), which provide services to each other and to other virtual roles according to the content of their resources and the related configuration information, through XML, Simple Object Access Protocol (SOAP) messages, and Web Service Description Language (WSDL). A software tool has been implemented to assist users to construct a nested P2PSLN.

To solve the second problem, this paper proposes an approach to measuring semantic similarity between peers. We consider not only semantic similarity between peers' data schemas, but also semantic similarity between structures in the schemas. Upon receiving a query, a peer will forward it only to relevant peers according to semantic link types as well as the similarity between nodes and the structures of peer schemas.

To solve the third problem, this paper proposes three mappings (mapping by *semantic node*, by *semantic clique*, and by *semantic path*) to reformulate a query from the source schema to target schemas. We use a Quality of Peers (QoP) method employing user-perceived qualities to manage inconsistent data in returned data flows.

Users first locate the required resources by finding a category in the Knowledge Grid and then browse the P2PSLN within the scope of that category. The P2PSLN reflects the explicit semantic relationships between a variety of resources. The combination of the Knowledge Grid and P2PSLN could have a great effect on the future interconnection environment. The results from theoretical analysis and from simulations demonstrate the effectiveness of our approach.

2. OVERVIEW

A P2PSLN is a directed network, where nodes are peers or P2PSLNs, and links specify semantic relations between peers. In a P2PSLN, each peer is an active and intelligent soft device that can dynamically and intelligently establish semantic connections with others.

A peer can be a server when providing data, information, or services, a router when forwarding queries, and a client when receiving data, information, or services from other peers.

As shown in Figure 1, each peer in a P2PSLN has two main modules: a *communication* module and a *data management* module. Peers communicate with each other using the SOAP messages. Users can query a peer through a graphical user interface (GUI) or by using SSeIQL—an SQL-like query language designed for peer data management in our P2PSLN.

The data management module of each peer is responsible for managing queries and answers. Upon receiving a query, the data management module performs the following tasks:

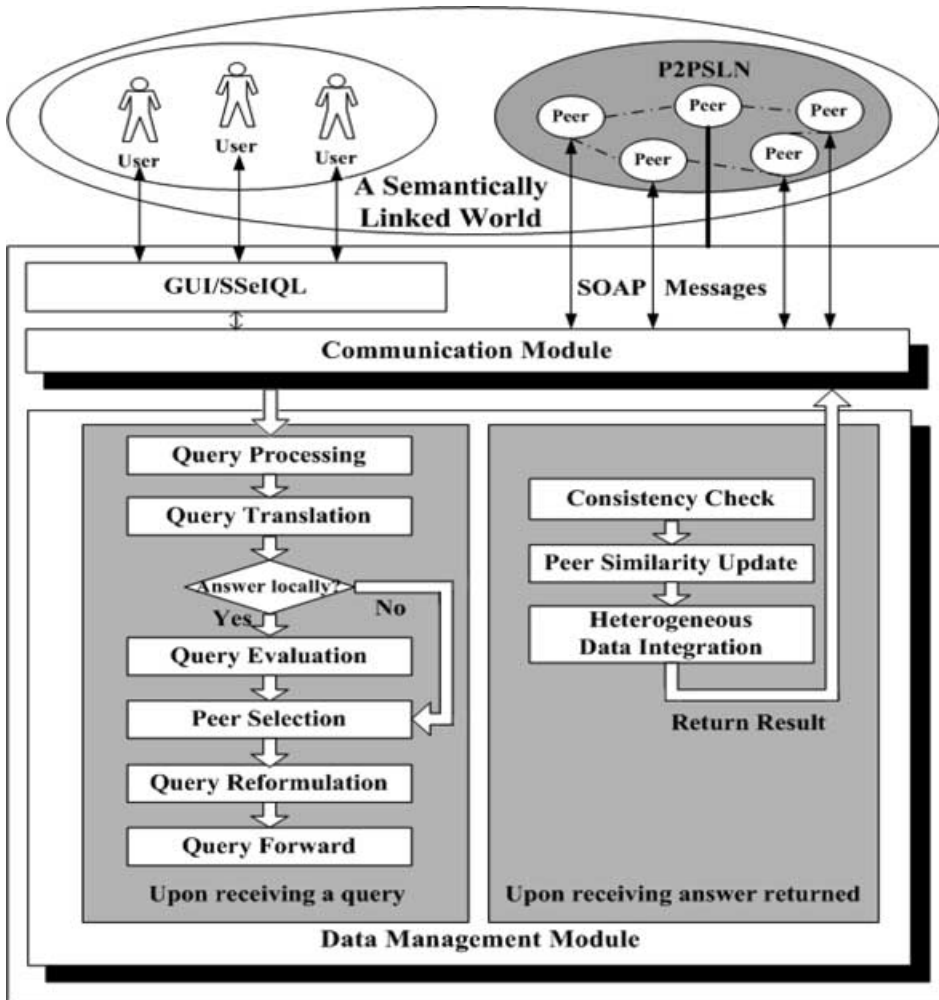


FIGURE 1. An overview of data management in a P2PSLN.

1. *Query Processing*. Analyzes the query and extracts its parameters.
2. *Query Translation*. Matches the query to the XML schema of the current peer to check whether it might be able to answer the query. If not, in step (6) the query is forwarded to the successors that are likely to be able to answer the query or to forward the query appropriately.
3. *Query Evaluation*. Puts the query to the current peer for answering.
4. *Peer Selection*. Selects the promising successors according to the semantic relationship and similarity between the current peer and possible successors.
5. *Query Reformulation*. Reformulates the query put to the current peer using the schemas of its immediate successors.
6. *Query Forwarding*. Autonomously forwards the query to the selected successors according to the routing policy and a preset time-to-live (TTL) value.

Upon receiving an answer from a successor, the data management module of the peer initiating the query analyzes the answer and looks for data inconsistent with the query. If a successor was sent a query but returned few answers, the current peer sends SOAP messages

to it to find out whether the successor still exists in the current P2PSLN and whether there is any schema change. If there is, it updates the schema mapping, semantic link type, and the similarity degree between them. Finally, the data management module combines the data in the answers to provide users or peers with a uniform view of the data from the sources.

3. THE P2PSLN MODEL

The P2PSLN is an extension of the P2P network. Semantic links represent the semantic relationships between peers and can be established according to two types of information: (1) peers' data schemas; and (2) the semantic description generated when P_i and P_j join a P2PSLN. Attaching semantic properties to the P2P network enables peers to efficiently and effectively route queries to appropriate successors.

3.1. Semantic Links

In a P2PSLN, a *semantic link* between two peers is a pointer with a type α directed from the predecessor peer to its successor. Depending on the relationship between the XML schemas of the peers and their descriptions, a *semantic link* can be one of the following types:

1. *Equal-to*, denoted by $P_i \text{---} equ \text{---} P_j$, says that P_i is semantically equal to P_j . The equal-to link is reflexive, symmetric, and transitive.
2. *Similar-to*, denoted by $P_i \text{---}(sim, sd) \text{---} P_j$, says that P_i is semantically similar to P_j to the degree sd .
3. *Reference*, denoted by $P_i \text{---} ref \text{---} P_j$, says that P_i refers semantically to P_j .
4. *Implication*, denoted by $P_i \text{---} imp \text{---} P_j$, says that P_i semantically implies P_j . The implication link is transitive and can help the reasoning mechanism to find new semantic implication relationships.
5. *Subtype*, denoted by $P_i \text{---} st \text{---} P_j$, says that P_j is semantically a part of P_i . The subtype link is transitive.
6. *Sequential*, denoted by $P_i \text{---} seq \text{---} P_j$, says that the content of P_j is the successor of the content of P_i in a context.
7. *Empty*, denoted by $P_i \text{---} \emptyset \text{---} P_j$, says that P_i and P_j are semantically unrelated.
8. *Null or Unknown*, denoted by $P_i \text{---} N \text{---} P_j$, says that no semantic relationship between P_i and P_j is known for certain.

Representing a P2PSLN as an adjacency matrix, the algebraic model and its characteristics are developed by Zhuge (2004).

Let $Schema(P_i)$ and $Schema(P_j)$ be the data schemas of P_i and P_j . Let $SD(P_i)$ and $SD(P_j)$ be the semantic descriptions of P_i and P_j . Semantic links between peers can be automatically discovered and derived under the following heuristic rules:

1. *Equal-to Link*. If $Schema(P_i)$ has the same elements and the same structure as $Schema(P_j)$, and $SD(P_i)$ and $SD(P_j)$ are the same, then $P_i \text{---} equ \text{---} P_j$ and $P_j \text{---} equ \text{---} P_i$.
2. *Similar-to Link*. If $Schema(P_i)$ and $Schema(P_j)$ overlap, then $P_i \text{---} sim \text{---} P_j$ and $P_j \text{---} sim \text{---} P_i$.
3. *Reference Link*. If $Schema(P_j)$ contains a substructure further explaining a leaf node N_i in $Schema(P_i)$ (that is, if the semantics of node N_i can be further explained by referring to $Schema(P_j)$), then $P_i \text{---} ref \text{---} P_j$.
4. *Implication Link*. If $Schema(P_i)$ is the same as $Schema(P_j)$ and $SD(P_i)$ implies $SD(P_j)$, then $P_i \text{---} imp \text{---} P_j$.

TABLE 1. Reasoning Rules for P2P Semantic Link Networks

No.	Rules
Rule 1	$P_i \text{---} equ \rightarrow P_i$
Rule 2	$P_i \text{---} equ \rightarrow P_j \Rightarrow P_j \text{---} equ \rightarrow P_i$
Rule 3	$P_i \text{---} equ \rightarrow P_j, P_j \text{---} equ \rightarrow P_k \Rightarrow P_i \text{---} equ \rightarrow P_k$
Rule 4	$P_i \text{---} equ \rightarrow P_j, P_j \text{---} \alpha \rightarrow P_k \Rightarrow P_i \text{---} \alpha \rightarrow P_k$
Rule 5	$P_i \text{---} imp \rightarrow P_j, P_j \text{---} imp \rightarrow P_k \Rightarrow P_i \text{---} imp \rightarrow P_k$
Rule 6	$P_i \text{---} st \rightarrow P_j, P_j \text{---} st \rightarrow P_k \Rightarrow P_i \text{---} st \rightarrow P_k$
Rule 7	$P_i \text{---} imp \rightarrow P_j, P_j \text{---} st \rightarrow P_k \Rightarrow P_i \text{---} imp \rightarrow P_k$
Rule 8	$P_i \text{---} imp \rightarrow P_j, P_j \text{---} ref \rightarrow P_k \Rightarrow P_i \text{---} ref \rightarrow P_k$
Rule 9	$P_i \text{---} st \rightarrow P_j, P_j \text{---} imp \rightarrow P_k \Rightarrow P_i \text{---} imp \rightarrow P_k$
Rule 10	$P_i \text{---} st \rightarrow P_j, P_j \text{---} ref \rightarrow P_k \Rightarrow P_i \text{---} ref \rightarrow P_k$ $P_i \text{---} st \rightarrow P_j, P_j \text{---} ins \rightarrow P_k \Rightarrow P_i \text{---} ins \rightarrow P_k$
Rule 11	$P_i \text{---} N \rightarrow P_j, P_j \text{---} \alpha \rightarrow P_k \Rightarrow P_i \text{---} N \rightarrow P_k$
Rule 12	$P_i \text{---} \emptyset \rightarrow P_j, P_j \text{---} \alpha \rightarrow P_k \Rightarrow P_i \text{---} N \rightarrow P_k$

5. *Subtype Link*. If *Schema* (P_j) is included in *Schema* (P_i), then $P_i \text{---} st \rightarrow P_j$.
6. *Sequential Link*. If *SD* (P_i) and *SD* (P_j) represent some sequential order (for example, determined by time), then $P_i \text{---} seq \rightarrow P_j$ can be established.
7. *Empty Link*. If P_i and P_j do not have any explicit semantic relationship, then $P_i \text{---} \emptyset \rightarrow P_j$ and $P_j \text{---} \emptyset \rightarrow P_i$.

We can chain semantic links to derive uncertain semantic relations between peers using reasoning rules (Zhuge 2003). The heuristic rules suitable for connecting different types of semantic links in a P2PSLN are listed in Table 1, where $\alpha \in \{equ, sim, ref, imp, st, seq, \emptyset, N\}$.

3.2. Operations on P2PSLNs

A P2PSLN supports three types of peer operations: *Join*, *Departure*, and *Stabilization*. Details of the various functions used are given in Table 2.

TABLE 2. Operations on P2P Semantic Link Networks

ID	Operation	Description
1	$P_i \text{..} Join (P2PSLN, P_j, \alpha)$	To make P_j the successor of P_i and specify the semantic link type α between P_i and P_j in a P2PSLN.
2	$P_i \text{..} FindSuccessors (P2PSLN, P_j, \alpha, P_k, \beta)$	To deduce semantic relationships between P_i and P_k in a P2PSLN provided that $P_i \text{---} \alpha \rightarrow P_j$ and $P_j \text{---} \beta \rightarrow P_k$ hold.
3	$P_i \text{..} SchemaInquiry (P2PSLN, P_j)$	To acquire the XML schema of P_j in a P2PSLN.
4	$P_i \text{..} Departure (P2PSLN)$	To leave a P2PSLN.
5	$P_i \text{..} Stabilization (P2PSLN, P_j)$	To ask for the existence and schema change from P_j in a P2PSLN.

1. *Peer Join*. When a peer P_i joins a P2PSLN, it will first identify how it relates to a randomly selected peer P_j in the network. Only when the semantic relationship satisfies some user-defined constraints can P_i take P_j as its immediate successor by issuing $P_i.Join(P2PSLN, P_j, \alpha)$, where α is semantic relationship between P_i and P_j . The semantic relationship between P_i and any other peer could be derived using the rules shown in Table 1. To find other successors, P_i will ask each successor P_k of P_j by issuing $P_i.FindSuccessors(P2PSLN, P_j, \alpha, P_k, \beta)$. If $P_i-\alpha \rightarrow P_j, P_j-\beta \rightarrow P_k \Rightarrow P_i-\gamma \rightarrow P_k$ satisfies the rules in Table 1, then P_i makes P_k its successor, and issues $P_i.FindSuccessors(P2PSLN, P_k, \gamma, P_m, \delta)$ iteratively within a preset TTL. After establishing the semantic relationships between P_i and its successors, P_i issues $P_i.SchemaInquiry(P2PSLN, P_j)$ to acquire the XML schemas of each successor P_j . The process to measure the similarity degree between peers with *Similar-to* link type is described in Section 5.
2. *Peer Departure*. Before a peer P_i leaves a P2PSLN, it will notify its predecessors and successors. In turn, predecessor P_j will remove P_i from its successor list, delete the semantic links between P_i and P_j , and add each successor P_k of P_i as its own successor provided that: (1) $P_k \notin P_j$'s successor list, and (2) there is a semantic relationship between P_j and P_k . Similarly, successor P_k will remove P_i from its predecessor list, delete the semantic links between them, and add each predecessor P_j of P_i as its own predecessor if: (1) $P_j \notin P_k$'s predecessor list, and (2) there is a semantic relationship between P_k and P_j .
3. *Peer Stabilization*. To ensure up-to-date semantic links, each P_i in a P2PSLN issues $P_i.Stabilization(P2PSLN, P_j)$ periodically in the background to have semantic link types and predecessor and successor pointers updated. If a predecessor or successor P_j exists in the network, it will notify P_i of its existence and of any schema change. Otherwise P_i will remove P_j from its predecessor or successor list and modify its neighbor index accordingly. When the XML schema of a peer changes, it will autonomously notify its predecessors and successors of the new schema through the SOAP messages.

3.3. P2PSLN Generation

Besides devising how to derive semantic links between peers automatically, we have developed a tool to help users construct and maintain a P2PSLN. There are two kinds of basic elements in a nested P2PSLN: nodes and semantic links. A node can be either a peer or a P2PSLN. A semantic link defines the relationship and similarity degree between two peer schemas. A graphical interface of the definition and maintenance tool is shown in Figure 2. Users can define a new P2PSLN or maintain an existing P2PSLN by clicking the operation buttons arranged in the top portion and drawing on the screen. Semantic constraints (that is, the user-defined constraints to be satisfied to take a peer as the neighbor of the current peer) will be considered while adding a new peer to an existing P2PSLN. The scalable-nested node hierarchy of the current P2PSLN is arranged in the left column. The description for each peer (*PeerID, Peer Name, Peer IP, Peer Description*) and each semantic link (*Predecessor, Successor, Semantic Relationship, Similarity Degree*) is listed at the bottom.

4. PEER SCHEMA MAPPING

Because the schemas between the source and the target peer may be quite different, schema mapping is proposed to solve the problem of inconsistency. Upon receiving peer schemas through SOAP messages, a peer will traverse the schemas recursively in depth-first order and extract node and path information from each target schema, and then carry out

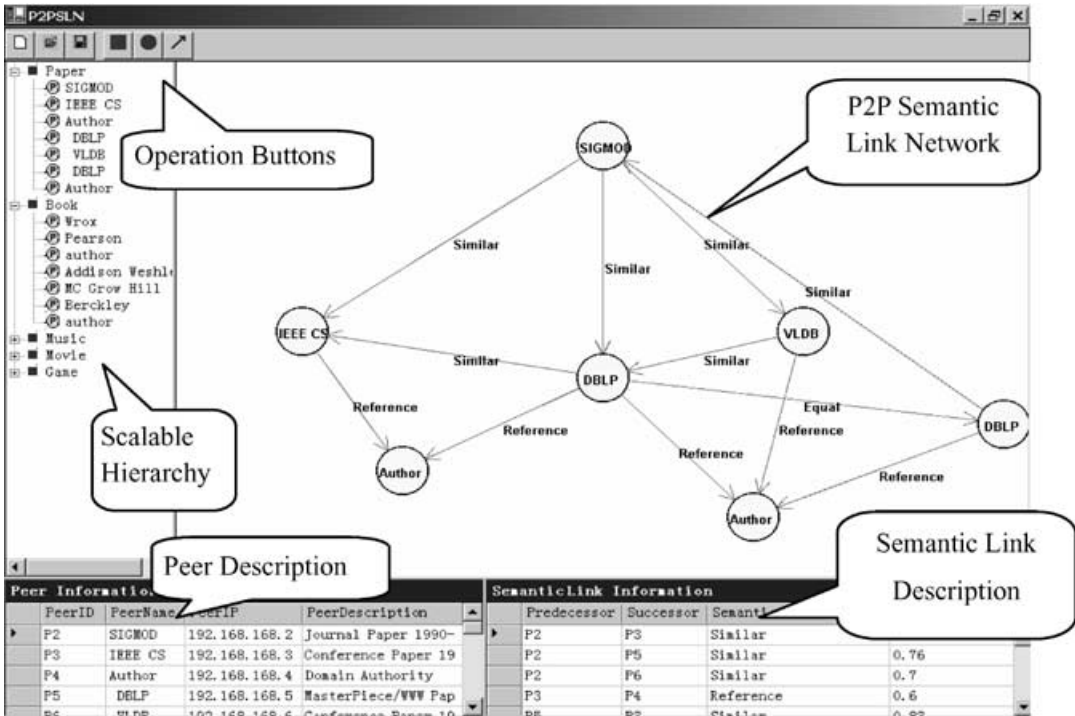


FIGURE 2. An interface of the proposed P2PSLN definition tool.

three types of mappings: semantic node mapping, semantic clique mapping, and semantic path mapping.

4.1. Semantic Node Mapping

Semantic node mapping resolves semantic inconsistency between nodes by mapping nodes in the source schema into nodes in target schemas. A peer encapsulates a global dictionary that defines a set of semantically related terms (synonymy, abbreviations, and so on) and the similarity degree between terms. The main challenge is to construct and maintain the global dictionary in a decentralized manner. We build up the global dictionary by using *WordNet*, and now flooding approach is applied in the case of update. The cost of the operation is $O(n)$, where n is the number of nodes in a P2PSLN.

After acquiring the target peer schemas, the source peer will automatically build mapping and similarity degree between nodes according to the definition in the global dictionary. The nodes in source schemas and the mapping nodes in target schemas are called *Semantic Nodes* and *Semantic Mapping Nodes*, respectively, when the semantic links have been established. We also provide tools to enable users to manually modify the semantic node mappings generated by the system, and to keep the new mappings in a local dictionary.

4.2. Semantic Clique Mapping

The semantic node mapping only maps elements of source schema into those of target schemas without considering the structure. The logic behind building semantic clique is to capture the semantic structure (that is, any parent-child or ancestor-descendant relationship

between a set of closely related nodes). The semantics of a node in a clique is constrained by the semantics of all nodes on the path from the root to the current node.

Semantic Clique Mapping identifies cliques (subtrees that cover a set of closely related nodes) and maps each *Semantic Clique (SC)* in a source schema into the target schemas, where the mapping images are called *Semantic Mapping Cliques (SMCs)*. Semantic mapping nodes in an *SMC* hold the structure that the nodes in a semantic clique have.

To find out the *SCs*, we first divide all the nodes in a source schema into a set of closely related sets, that is, the semantic node sets. The following algorithm identifies the *Semantic Cliques* and *Semantic Mapping Cliques* corresponding to each node set.

Algorithm SemanticCliqueRecognition (T_1, T_2, SN)

/* Given a set of closely related semantic nodes $SN = \{SN_1, \dots, SN_n\}$, find semantic cliques in sub-tree rooted at T_1 and semantic mapping cliques in sub-tree rooted at T_2 */

Input: $T_1, T_2, SN = \{SN_1, \dots, SN_n\}$

Output: $SC = \{SC_1, \dots, SC_k\}, SMC = \{SMC_1, \dots, SMC_k\}$ /* the semantic clique set in the source schema and the semantic mapping clique set in the target schemas */

Begin

IF ($T_1 == \text{Null}$)

THEN Return True;

$R_1 = T_1.\text{FirstChild}$; Temp = True;

WHILE ($R_1 \neq \text{NULL}$)

$R_2 = \text{Semantic-Mapping-Node}(T_2, R_1)$;

IF ($R_2 == \text{Null}$)

THEN Return False;

ELSE

Temp = Temp and *SemanticCliqueRecognition* (R_1, R_2, SN);

IF Temp == False

THEN Return False;

ELSE

Add R_1 To *SC*; /* add R_1 to semantic clique set */

Add R_2 To *SMC*; /* add R_2 to semantic mapping clique set */

$R_1 = T_1.\text{NextChild}$;

END IF;

END IF;

END WHILE;

Return Temp;

END IF;

End

The Maximum Semantic Clique is the *SC* that is not semantically included by any other *SC*. Any node in an *SC* does not have predecessor–descendant relations with nodes in other *SCs*. *The Minimum Common Sub-tree* denoted by *MCS* ($SC_1, \dots, SC_p, SN_1, \dots, SN_q$) is the subtree that covers all the *SCs* (SC_1, \dots, SC_p) and all the identified nodes (SN_1, \dots, SN_q) not belonging to any *SC* in a source schema. The root of the minimum common subtree is called the *Nearest Common Predecessor* of the involved *SCs* and nodes. Algorithms to find the minimum common subtree could be found in Lu (1984).

We have also developed a tool to assist users to define *SCs*. Figure 3 depicts the user interface for defining *SCs* and *SMCs*. The middle-left-hand portion displays the source schema

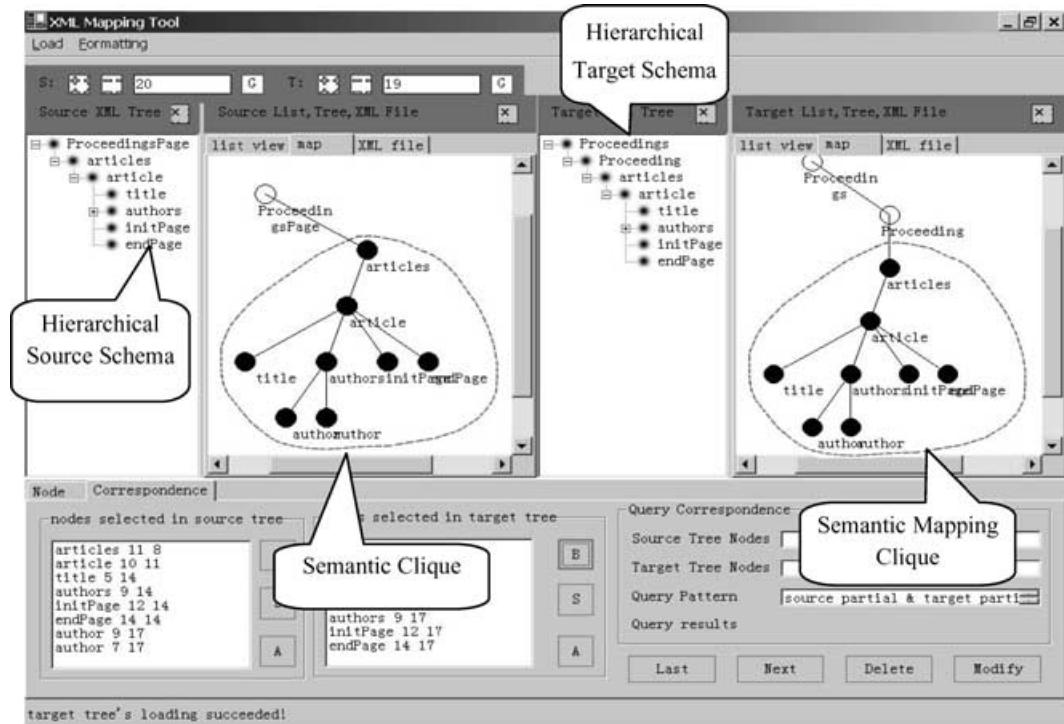


FIGURE 3. User interface to define SCs and SMCs.

hierarchy and its graphical representation, while the middle-right-hand portion corresponds to the target schema. The black nodes in the source schema form a user-defined SC, while the black nodes in the target schema are the corresponding SMC.

Figure 4 depicts XML trees conforming to schemas in the ACM SIGMOD and VLDB proceedings. The identified semantic nodes and semantic mapping nodes are the marked circles. The SCs, the maximum SC, and the minimum common subtree are marked by dashed lines as the legend explains.

4.3. Semantic Path Mapping

Semantic Path Mapping maps each path from the root to the nodes in a source schema into semantic mapping paths in target schemas. Let $Schema(P_i)$ be the data schema of P_i , $Semantic-Path(N_i)$ be the path from $Root(N_i)$ to node N_i in a source schema, and $Semantic-Mapping-Path(N_i)$ be the mapping path of $Semantic-Path(N_i)$ in target schemas. The process of semantic path mapping is as follows.

Algorithm SemanticMappingPath ($Schema(P_i), N_i$)

Input: $Schema(P_i)$ /* Schema of P_i */;

N_i /* Semantic Node in $Schema(P_i)$ */;

Output: $Semantic-Mapping-Path(N_i)$;

Step 1: For each node on $Semantic-Path(N_i)$

Find semantic mapping nodes in target schemas;

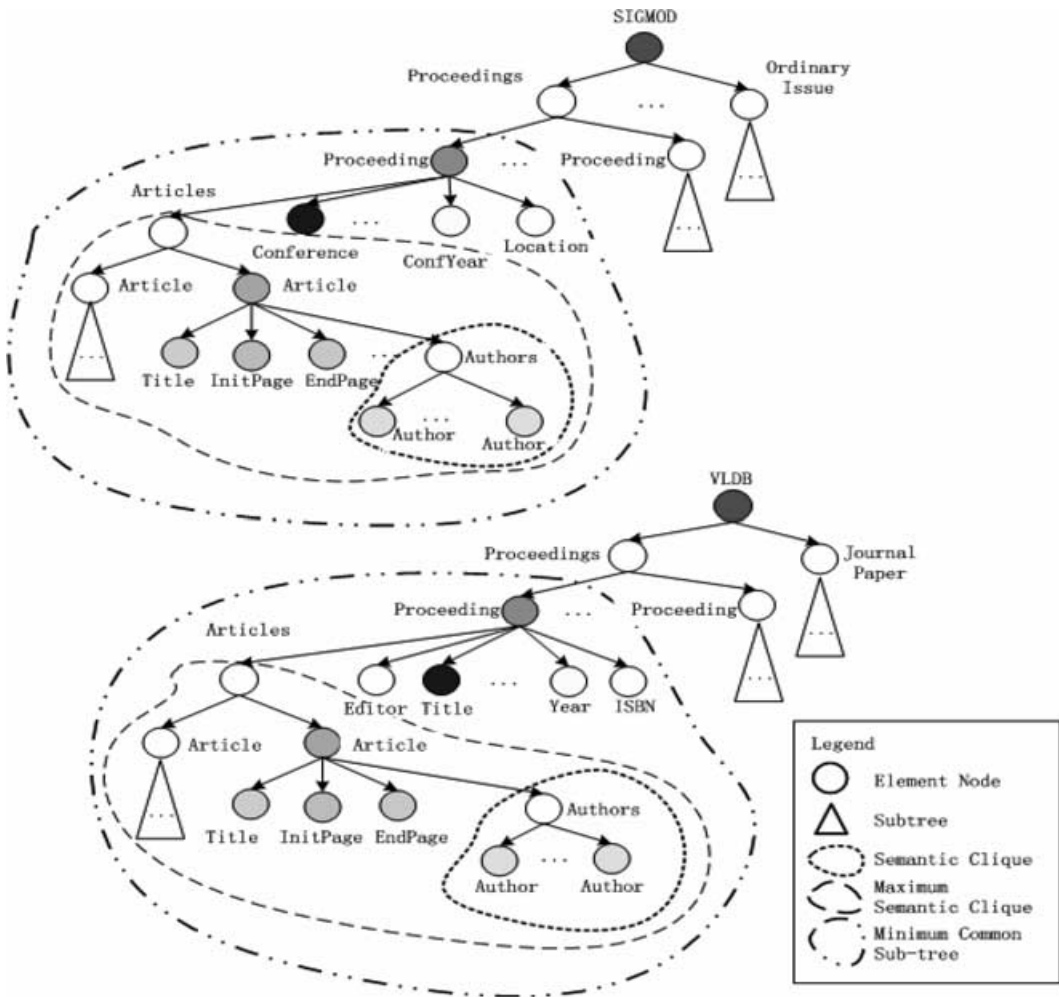


FIGURE 4. XML trees of ACM SIGMOD and VLDB proceedings.

Step 2: Connect semantic mapping nodes in target schemas to form an identified path;
 Step 3: IF the identified path matches a path *SMP*ath in target schemas
 THEN
 Return (*SMP*ath);
 ELSE
 Extend the identified path by replacing parent–child relations with
 ancestor–descendant relations between adjacent nodes;
 IF path *SMP*ath in target schema contains the extended identified path
 THEN Return (*SMP*ath);
 END IF.

Based on this algorithm, Tables 3–5 show in turn the semantic node, semantic clique, and semantic path mappings of the schemas in Figure 4. Table 3 is based on the definitions in the global and local dictionaries, Table 4 on the *SemanticCliqueRecognition* algorithm, and Table 5 on the *SemanticMappingPath* algorithm.

TABLE 3. Semantic Node Mapping of SIGMOD and VLDB Proceedings

Source	Semantic Node	Target	Semantic Mapping Node	Similarity Degree (SD)
SIGMOD	SIGMOD	VLDB	VLDB	0.9
SIGMOD	Title	VLDB	Title	1
SIGMOD	InitPage	VLDB	InitPage	1
SIGMOD	EndPage	VLDB	EndPage	1
SIGMOD	Author	VLDB	Author	1
SIGMOD	Conference	VLDB	Title	0.5
SIGMOD	ConfYear	VLDB	Year	0.9
SIGMOD	...	VLDB

TABLE 4. Semantic Clique Mapping of SIGMOD and VLDB Proceedings

Source	Semantic Clique	Target	Semantic Mapping Clique	SD
SIGMOD	Authors (Author, ..., Author)	VLDB	Authors (Author, ..., Author)	1
SIGMOD	Articles (Title, InitPage, EndPage, Authors (Author, ..., Author))	VLDB	Articles (Title, InitPage, EndPage, Authors (Author, ..., Author))	1
SIGMOD	...	VLDB

TABLE 5. Semantic Path Mapping of SIGMOD and VLDB Proceedings

Source	Semantic Path	Target	Semantic Mapping Path	SD
SIGMOD	SIGMOD/Proceedings/Proceeding/Articles/Article/Title	VLDB	VLDB/Proceedings/Proceeding/Articles/Article/Title	1
SIGMOD	SIGMOD/Proceedings/Proceeding/Conference	VLDB	VLDB/Proceedings/Proceeding/Title	0.7
SIGMOD	...	VLDB

5. QUERY ROUTING IN P2PSLN

5.1. Peer Selection and Peer Similarity Measurement

A peer answers queries with the help of semantically relevant peers. For *similar-to* successors, the similarity degree will be measured. The similarity between two peers depends not only on the similarity between nodes but also on the similarity between semantic structures such as the parent–child relationship between the closely related semantic nodes in a maximum SC or a minimum common subtree.

Two basic variables for an algorithm determining the similarity between semantic structures are:

1. Leaf nodes' semantic contribution relevant to a query, because queries are of leaf nodes.
2. The contribution of ancestor nodes' semantic similarities, larger similarities make greater contributions.

The following notations are used in the algorithm:

- $Peer(N_i)$ is the semantic mapping node for node N_i .
- $Length(N_i, N_j)$ is the number of nodes on the path from N_i to N_j .
- $MaxSC(N_i)$ is the maximum SC including node N_i .
- $MinCS(N_i)$ is the minimum common sub-tree including node N_i .
- $Semantic-Node-SD(N_i, N_j)$ is the similarity degree between N_i and N_j .

The following algorithm is for measuring the structural similarity between N_i in the source schema and its mapping node N_j in the target schema.

Input: N_i, N_j /* N_i is a semantic node, and $N_j = Peer(N_i)$ */

Output: $Semantic-Structure-SD(N_i, N_j)$ /* the semantic structure similarity between N_i and N_j */

Step 1: IF N_i belongs to one of the *Maximum Semantic Cliques*

THEN $T = MaxSC(N_i)$

ELSE $T = MinCS(N_i)$.

END IF

Step 2: $Root(N_i) = T$

IF $Length(N_i, T) = 1$

THEN $Semantic-Structure-SD(N_i, N_j) = Semantic-Node-SD(N_i, N_j)$

ELSE

$NodeSet = \{N_i, \dots, Root(N_i)\}$ /* Nodes on path from N_i to $Root(N_i)$ */

$\vec{FV} = (fv_{N_i}, \dots, fv_{Root(N_i)})$ /* The semantic structure similarity feature vector for each node in $NodeSet$ */

$$fv_{N_k} = \begin{cases} 0, & Peer(N_k) \notin Semantic - Mapping - Path(N_i), \\ Semantic - Node - SD(N_k, Peer(N_k)), & Otherwise \end{cases} \quad (1)$$

$\vec{W} = (W_{N_i}, \dots, W_{Root(N_i)})$ /* \vec{W} is the weight vector to denote node importance for each node in $NodeSet$ */

$$W_{N_k} = \begin{cases} 1/2, & N_k = N_i \\ (1/2)^k, & k = length(N_i, N_k), \text{ and } N_k \neq Root(N_i) \\ 1 - \sum_{l=1}^{n-1} W_{N_l} = (1/2)^{n-1}, & n = length(N_i, Root(N_i)), N_k = Root(N_i) \end{cases} \quad (2)$$

$$Semantic - Structure - SD(N_i, N_j) = \frac{\vec{W} \cdot \vec{FV}}{|\vec{W}| |\vec{FV}|}, \quad (3)$$

where $\vec{W} \cdot \vec{FV} = W_{N_i}fv_{N_i} + \dots + W_{Root(N_i)}fv_{Root(N_i)}$,

and $|\vec{X}| = \|\vec{X}\|_2 = \sqrt{x_1^2 + \dots + x_k^2}$

END IF

Let $SN = \{N_1, \dots, N_m\}$ be the node set of the source schema, $\overrightarrow{SD} = (SR_{N_1}, \dots, SR_{N_m})$ be the feature vector for the structural similarity of each node calculated from formula (3), $\overrightarrow{W} = (W_{N_1}, \dots, W_{N_m})$ be the user-defined weight vector representing the importance of each node. The structural similarity between the two peer schemas is defined as follows:

$$Semantic - Structure - SD(P_i, P_j) = \frac{\overrightarrow{W} \cdot \overrightarrow{SD}}{\left| \overrightarrow{W} \right| \left| \overrightarrow{SD} \right|}. \quad (4)$$

5.2. Query Reformulation and Heterogeneous Data Integration

After selecting proper successors for query forwarding, the peer initiating the query will reformulate the query for its immediate successors, then for the successors' immediate successors, and so on. Whenever a forwarded query reaches a peer that holds the matching data, it will be processed and the result will be returned to the initiator. The semantic node, semantic clique, and semantic path mappings in Tables 3–5 are used for reformulating a query over target schemas.

Within a preset timeout, the peer initiating the query will analyze the returned data. To solve the problem of data inconsistency, we take into account the QoP, the user-perceived qualities such as the number of returned results, response time, traffic overhead, precision, recall, and so on. The data returned by peers with higher QoP are considered more likely to be consistent. Finally, the peer initiating the query will combine relevant data and then give a uniform view of the results to users and peers.

6. EXPERIMENTS AND DISCUSSION

6.1. The P2PSLN Simulation

To illustrate and evaluate the proposed approach, we have simulated a small but realistic P2PSLN of 50 peers. In the simulation, each peer randomly selects an average of six peers as its neighbors. The metadata of 500,000 papers is collected from DBLP XML databases (<http://dblp.uni-trier.de/xml/>) and ACM SIGMOD XML records (<http://www.acm.org/sigmod/record/xml/SigmodRecord/SigmodRecord.xml>). We have developed an algorithm to generate heterogeneous data schemas and distribute the metadata uniformly over all the peers. The XML document size of each peer varies from 275 to 14,207 KB. Each peer has the same bandwidth and process ability. Twenty randomly generated keyword-based queries are randomly submitted at random intervals to 20 peers selected randomly to test the performance of the P2PSLN compared with the following two types of routing methods:

1. The *Breadth First Search* (BFS): each peer broadcasts the query to all its neighbors.
2. The *Random Walk Search* (RW): each peer forwards the query to a number of randomly selected neighbors.

Our evaluation metrics are the *recall rates* (that is, the fraction of the relevant data that has been retrieved), and the *bandwidth consumption* (that is, the number of messages per query).

The graphical interface of the P2PSLN simulation is shown in Figure 5. The process for collecting experimental data includes the following interaction steps:

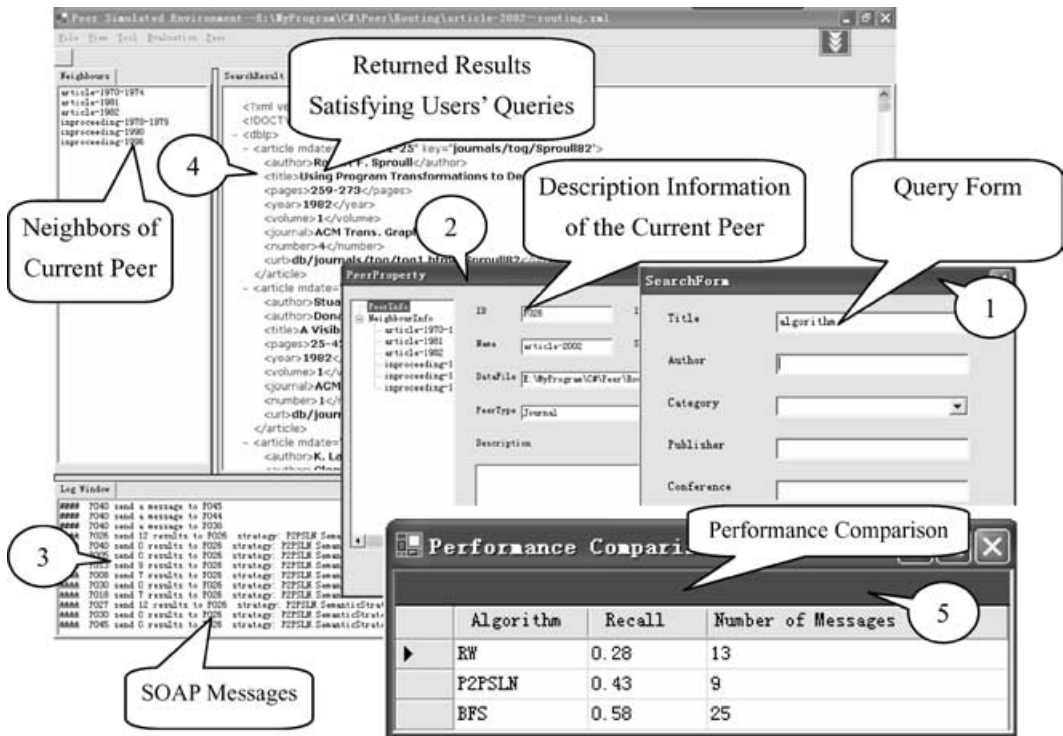


FIGURE 5. The interface of the P2PSLN simulation.

1. Users put in queries for the peers randomly selected from a list by filling in the query form in interface (1). The queries about the title and the author in the evaluation are keyword-based, while those related to publishing dates are for ranges of years.
2. The system displays the description of the current peer in interface (2).
3. SOAP messages generated during query routing are displayed in interface (3).
4. The returned results satisfying users' queries are shown in interface (4).
5. The system displays the comparison between the three routing policies in interface (5).

6.2. Static Performance Analysis

Our experiments evaluate the performance of the P2PSLN when no peer joins or leaves the network during the simulation.

The recall rates of the three routing policies from the first experiment, for a TTL of 5, are given in Figure 6. The average rates of BFS, P2PSLN, and RW are 0.58, 0.43, and 0.28, which shows the BFS routing policy achieving the best recall rate. This is because BFS broadcasts the queries to all its neighbors and it is sure to get most of the relevant data. P2PSLN forwards queries according to the semantic relationship and the similarity degree, thus it can get a higher recall rate than the RW routing policy within a preset TTL.

The second experiment counts the messages that the three routing policies generate within a preset TTL to process 20 queries. Figure 7 shows that the BFS generates the most messages, 25 on average. The P2PSLN and RW generate 9 and 13 on average. The BFS needs nearly three times as many messages as the P2PSLN. In P2PSLN searching, each peer in the

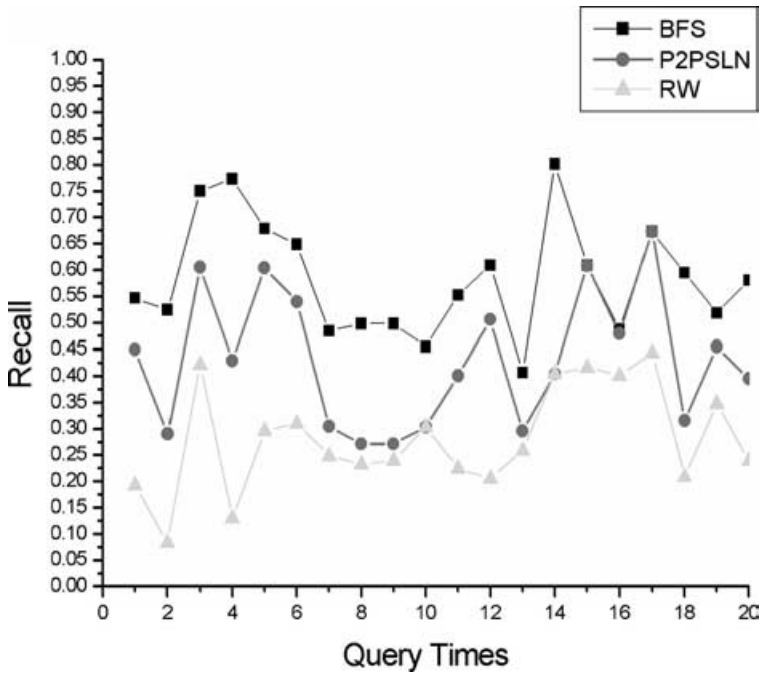


FIGURE 6. Recall rates for 20 queries for BFS, P2PSLN, and RW routing (TTL = 5).

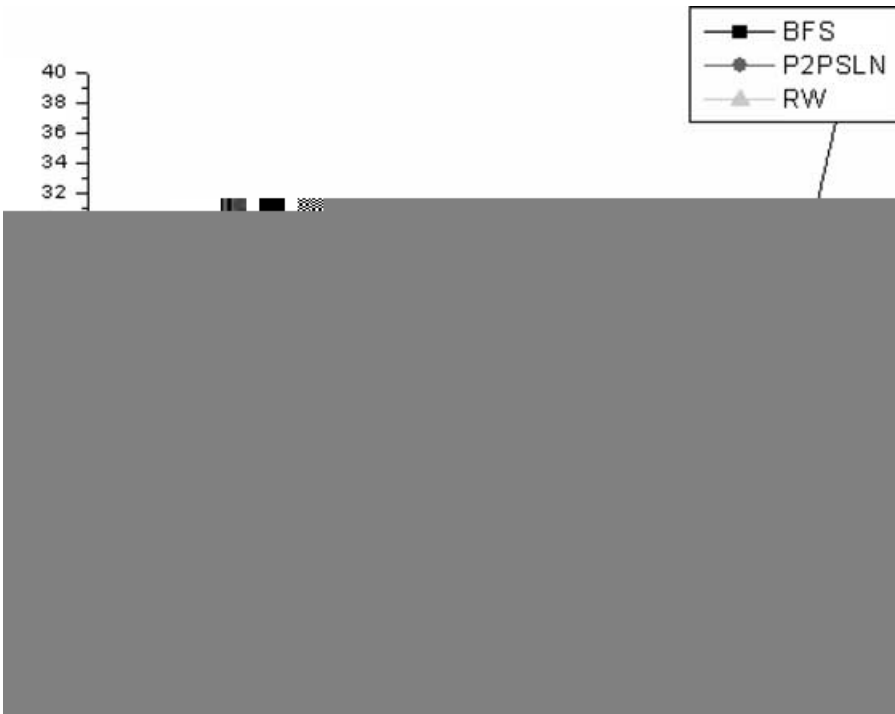


FIGURE 7. Number of messages generated by 20 queries in BFS, P2PSLN, and RW routing policies (TTL = 5).

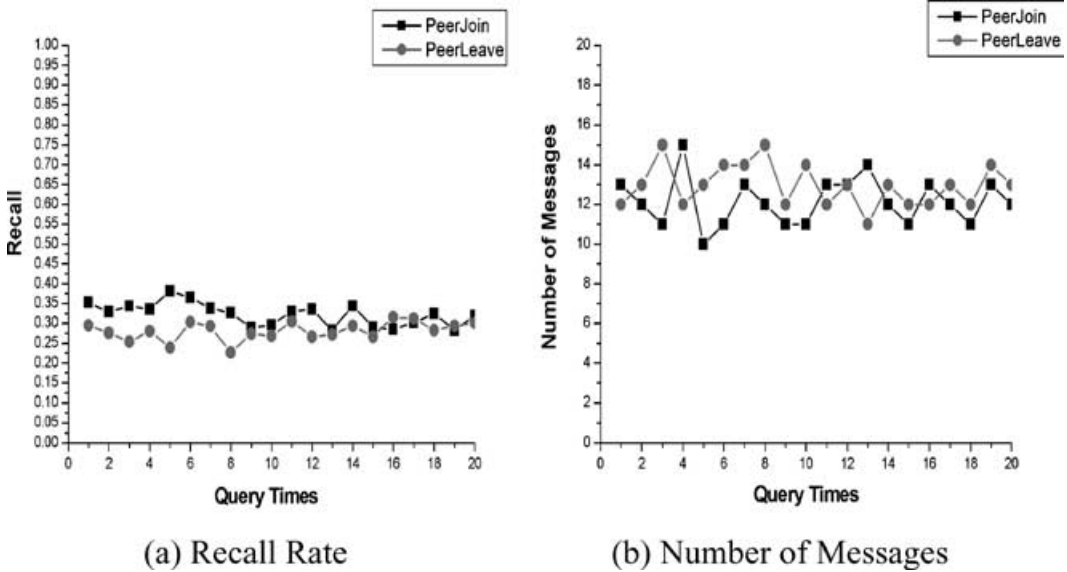


FIGURE 8. Performance of P2PSLN under dynamic environment (TTL = 5).

query path determines the similarity of its neighbors and sends the query only to the three most similar neighbors, which is why so few messages are needed.

6.3. Dynamic Performance Analysis

In the following two experiments, 20 queries were used to test the performance of a dynamic P2PSLN, that is, peers can join and leave at any time. Figure 8(a) plots the recall rate while 10% of the peers randomly join or leave the network. On average, the recall rates were 0.3228 with peers joining and 0.2812 with peers leaving. Figure 8(b) plots the number of messages required to answer a query. On average, the number of messages was 12 with peers joining and 13 with them leaving. As expected, the performance under dynamic conditions is not as good as that under static conditions.

6.4. Comparison and Discussion

The simulation results allow the effectiveness and efficiency of P2PSLN to be compared to the BFS and RW routing. Our approach is based on unstructured P2P networks, thus in the following we compare P2PSLN to other indexing approaches for unstructured P2P networks.

“Routing indices” (RIs) are based on document topics (Crespo and Garcia-Molina 2002). Simulations show these to improve performance. However, the topics of documents dealing with multiple domains may be unknown to developers of an index. Besides, improper classification will worsen query routing.

The approaches of cycle analysis and functional dependency analysis are to achieve semantic agreement between nodes (Aberer et al. 2003). However, its similarity measurement does not consider the semantic structure of an XML document. Actually, the semantics of a node in an XML document depends at least partly on the semantics of all nodes on the path from the root to the current node.

Multilevel breadth and depth bloom filters are used to maintain information about the structure of the XML documents (Koloniari and Pitoura 2004). Similarity between nodes is

related to the similarity between their document filters. Experiments show that this approach outperforms the classical Bloom filters in routing path queries.

Unlike these previous approaches, our approach uses semantic links to denote relationships between peer schemas. Each peer in a P2PSLN autonomously identifies semantic relationships between itself and its neighbors and interacts with them. The approach also includes an automatic semantic link discovery approach and a tool for building and maintaining P2PSLN. More semantic links can be derived through reasoning. The semantic clique captures the semantic structure (that is, any parent–child or ancestor–descendant relationship between a set of closely related nodes). The ancestor nodes' semantic similarities contribute to the semantics of the current node, larger similarities making greater contributions. The semantic-based peer similarity measurement provides a way to measure the similarity between a set of closely related nodes in peer schemas for efficient query routing.

Our approach also outperforms the previous routing index approach by providing users and other peers with a uniform view of results from different peers in a P2PSLN. The semantic node, clique, and path mappings resolve the issues of semantic and structural heterogeneity between source and target schemas. The issue of data inconsistency in the returned data flows is resolved on the basis of the quality of the peers involved.

7. CONCLUSIONS

A peer-to-peer system is a natural and extensible underlying layer for Knowledge Grid because of its autonomy, self-organization, and scalability. As a solution to query answering at the semantic overlay of the scalable Knowledge Grid, this paper proposes: a P2PSLN, a semantic-based peer similarity measurement for query routing, and a peer schema mapping approach for query reformulation. The results from theoretical analysis and simulations show that the proposed approach is effective.

The significant contributions of this work are: (1) the notion of a P2PSLN, a method of automatically discovering semantic links, and a tool for constructing and maintaining a nested P2PSLN; (2) the use of semantic structure similarity to measure the similarity between peers to improve the effectiveness and efficiency of query routing; and (3) an approach for uniformly providing users and peers with data obtained from multiple peers that eliminates semantic isolation, heterogeneity, and inconsistency. Experiments show that the proposed approach is promising for peer data management in the P2PSLN.

The approach has been implemented for, and integrated into, the China e-science Knowledge Grid environment, which aims to provide access to distributed resources (including information, knowledge, and services) and speed up the processes of knowledge generation, propagation, fusion, and management in cooperative research.

Ongoing work includes three aspects: (1) semantically clustering relevant peers to form a hierarchical P2PSLN to improve the performance of the proposed P2PSLN model; (2) incorporating query reformulation optimization into the proposed approach; and, (3) incorporating our approach with others, such as those indexing on actual data to improve the effectiveness of query routing.

ACKNOWLEDGMENTS

The authors thank all members of the China Knowledge Grid Research Group (<http://kg.ict.ac.cn>) for their help and cooperation. Thanks also go to the anonymous reviewers and the guest editors for their valuable comments.

REFERENCES

- ABERER, K., P. CUDRE-MAUROUX, and M. HAUSWIRTH. 2003. The chatty Web: Emergent semantics through gossiping. *In Proceedings of the 12th International World Wide Web Conference (WWW2003)*. Association for Computing Machinery, Budapest, Hungary, pp. 197–206.
- BALAKRISHNAN, H., M. F. KAASHOEK, D. KARGER, R. MORRIS, and I. STOICA. 2003. Looking up data in P2P systems. *Communications of the ACM*, **46**(2):43–48.
- BERNSTEIN, P., F. GIUNCHIGLIA, A. KEMENTSIETSIDIS, J. MYLOPOULOS, L. SERAFINI, and I. ZAIHRAYEU. 2002. Data management for peer-to-peer computing: A vision. *In Proceedings of Workshop on the Web and Databases (WebDB 2002)*, Madison, Wisconsin, pp. 89–94.
- BROEKSTRA, J., M. EHRIG, P. HAASE, F. HARMELEN, M. MENKEN, P. MIKA, B. SCHNIZLER, and R. SIEBES. 2004. Bibster—A semantics-based bibliographic peer-to-peer systems. *In Proceedings of the Second Workshop on Semantics in Peer-to-Peer and Grid Computing*, New York, USA, pp. 3–22.
- CLARKE, I., O. SANDBERG, B. WILEY, and T. HONG. 2000. Freenet: A distributed anonymous information storage and retrieval system. *In Proceedings of the ICSI Workshop on Design Issues in Anonymity and Unobservability*, Berkeley, CA, pp. 311–320.
- CRESPO, A., and H. GARCIA-MOLINA. 2002. Routing indices for peer-to-peer systems. *In Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS 2002)*, Vienna, Austria, pp. 23–32.
- DEUTSCH, A., and V. TANNEN. 2003. MARS: A system for publishing XML from mixed and redundant storage. *In Proceedings of the 29th VLDB Conference (VLDB 2003)*, Berlin, Germany, pp. 201–212.
- GARCES-ERICE, L., P. A. FELBER, E. W. BIRSACK, G. URVOY-KELLER, and K. W. ROSS. 2004. Data indexing in peer-to-peer DHT networks. *In Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS 2004)*, Tokyo, Japan, pp. 200–208.
- GIUNCHIGLIA, F., and I. ZAIHRAYEU. 2002. Making peer databases interact—A vision for an architecture supporting data coordination. *In Proceedings of the Conference on Information Agents (CIA 2002)*, Madrid, Spain, pp. 18–35.
- HALEVY, A. Y., Z. IVES, D. SUCIU, and I. TATARINOV. 2003a. Schema mediation in peer data management systems. *In Proceedings of the 19th International Conference on Data Engineering (ICDE 2003)*, Bangalore, India, pp. 505–518.
- HALEVY, A. Y., Z. IVES, P. MORK, and I. TATARINOV. 2003b. Piazza: Data management infrastructure for semantic web applications. *In Proceedings of the 12th International World Wide Web Conference (WWW2003)*. Association for Computing Machinery, Budapest, Hungary, pp. 556–567.
- HARREN, M., J. M. HELLERSTEIN, R. HUEBSCH, B. LOO, S. SHENKER, and I. STOICA. 2002. Complex queries in DHT-based peer-to-peer networks. *In Proceedings of IPTPS02*, Cambridge, USA, pp. 242–259.
- KEMENTSIETSIDIS, A., M. ARENAS, and R. MILLER. 2003. Mapping data in peer-to-peer systems: Semantics and algorithmic issues. *In Proceedings of the ACM SIGMOD 2003*, San Diego, California, pp. 325–336.
- KOLONIARI, G., and E. PITOURA. 2004. Content-based routing of path queries in peer-to-peer systems. *In Proceedings of the Ninth International Conference on Extending Database Technology (EDBT 2004)*, Heraklion-Crete, Greece, pp. 29–47.
- LI, M., W. LEE, A. SIVASUBRAMANIAM, and D. LEE. 2004. A small world overlay network for semantic based search in P2P systems. *In Proceedings of the Second Workshop on Semantics in Peer-to-Peer and Grid Computing*, New York, USA, pp. 71–90.
- LÖSER, A., F. NAUMANN, W. SIBERSKI, W. NEJDL, and U. THADEN. 2003. Semantic overlay clusters within supper-peer networks. *In Proceedings of International Workshop on Databases, Information Systems, and P2P Computing (DBISP2P 2003)*. Lecture Notes in Computer Science, Springer Verlag, Berlin, Germany, pp. 33–47.
- LU, S. Y. 1984. A tree-matching algorithm based on node splitting and merging. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **6**(2):249–256.

- MADHAVAN, J., P. BERNSTEIN, and E. RAHM. 2001. Generic schema matching with Cupid. *In Proceedings of the 27th VLDB Conference (VLDB 2001)*, Roma, Italy, pp. 49–58.
- NAKAUCHI, K., H. MORIKAWA, and T. AOYAMA. 2004. Semantic peer-to-peer search using query expansion. *In Proceedings of the Second Workshop on Semantics in Peer-to-Peer and Grid Computing (SemPGRID 2004)*, New York, USA.
- NEJDL, W., M. WOLPERS, W. SIBERSKI, and C. SCHMITZ. 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 (WWW2003)*. Association for Computing Machinery, Budapest, Hungary, pp. 536–543.
- NG, W. S., B. C. OOI, K. TAN, and A. ZHOU. 2003. PeerDB: A P2P-based system for distributed data sharing. *In Proceedings of International Conference on Data Engineering (ICDE 2003)*, Bangalore, India, pp. 633–644.
- OOI, B., Y. SHU, and K. TAN. 2003. DB-enabled peers for managing distributed data. *In Proceedings of the Fifth Asia-Pacific Web Conference (APWeb 2003)*, Xi'an, China, pp. 10–21.
- RAHM, E., and P. BERNSTEIN. 2001. A survey of approaches to automatic schema matching. *VLDB Journal*, **10**(4):334–350.
- RATNASAMY, S., P. FRANCIS, M. HANDLEY, R. KARP, and S. SHENKER. 2001. A scalable content-addressable network. *In Proceedings of ACM SIGCOMM*, San Diego, CA, pp. 161–172.
- RATNASAMY, S., S. SHENKER, and I. STOICA. 2002. Routing algorithms for DHTs: Some open questions. *In Proceedings of the First International Workshop on Peer-to-Peer Systems (IPTPS'02)*, Cambridge, USA, pp. 45–52.
- ROWSTRON, A., and P. DRUSCHEL. 2001. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. *In Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms*, Heidelberg, Germany, pp. 329–350.
- STOICA, I., R. MORRIS, D. KARGER, M. KAASHOEK, and H. BALAKRISHNAN. 2001. Chord: A scalable peer-to-peer lookup service for internet applications. *In Proceedings of ACM SIGCOMM 2001*, San Diego, CA, USA, pp. 149–160.
- TAND, C., Z. XU, and M. MAHALINGAM. 2003. pSearch: Information retrieval in structured overlays. *ACM SIGCOMM Computer Communications Review*, **33**(1):89–94.
- TATARINOV, I., and A. HALEVY. 2004. Efficient query reformulation in peer-data management systems. *In Proceedings of ACM SIGMOD 2004*, Paris, France, pp. 539–550.
- TIWANA, A. 2003. Affinity to infinity in peer-to-peer knowledge platforms. *Communications of the ACM*, **46**(5):77–80.
- TSOUMAKOS, D., and N. ROUSSOPOULOS. 2003. A comparison of peer-to-peer search methods. *In Proceedings of Sixth International Workshop on the Web and Databases*, San Diego, USA, pp. 61–66.
- WULFF, M., and G. SAKARYAN. 2004. Search methods in P2P networks: A survey. *In Proceedings of I2CS-Innovative Internet Community Systems (I2CS 2004)*. Lecture Notes in Computer Science, Springer Verlag, Guadalajara, Mexico.
- ZHUGE, H. 2002. Clustering soft-devices in Semantic Grid. *Computing in Science and Engineering*, **4**(6):60–63.
- ZHUGE, H. 2003. Active e-document framework ADF: Model and tool. *Information and Management*, **41**(1):87–97.
- ZHUGE, H. 2004. *The Knowledge Grid*. World Scientific Publishing Co., 280 p., Singapore.