Ontology-based Peer Exchange Network(OPEN)*

Hui Dong, Zhimin Wang Robert A. Morris, and Jun Huang

University of Massachusetts Boston, 100 Morrissey Blvd Boston MA 02125, USA
hdong1, wangzm, ram@cs.umb.edu

Abstract. Peer-to-peer systems represent a widely accepted approach to sharing massive data and services among large, diverse and varying sets of nodes in the network. In this paper, we introduce an overlay network which constructs the logical network topologies using a global ontology and peer ontological characteristics. Network construction and query model are defined to allow the efficient answer of complex queries based on concepts and relations. Simulation results are used to verify the predicted network properties.

1 Introduction

In work on collaborative image annotation [16], we are developing a system whereby human or software agents anywhere in the internet can jointly edit shared objects by the use of editors which can exchange messages about the semantics of the editing enterprise. The message passing infrastructure of that project is a general purpose P2P network described in this paper in which queries are routed efficiently by the use of semantic information published by each peer about the nature of its data and services. In the case of the image annotation project, that information includes what kind of images the peer can store and retrieve and what image annotation messages it can process.

Digitalized data produced worldwide have exceeded exabyte scale and grow much faster every year [1]. With the success of distributed file sharing systems like Napster [2] and Gnutella [3], peer-to-peer (P2P) systems have gained popularity because of their decentralized structure and self-organized nature. To make P2P systems capable of responding to large numbers of user queries simultaneously and scale well with potentially exponential information growth, the mechanism of efficiently routing query messages to peers which hold answers has brought intensive studies. Depending on the overlay network topologies, such systems are usually divided into three categories: unstructured, structured, and hybrid networks. Unstructured networks usually have a random topology and route messages using flooding or random walk [4]. This approach may send queries to many irrelevant intermediate nodes, and hence yields a long response time and wastes network bandwidth. CAN [5] and CHORD [6] have proposed a rigid network structure based on distributed hash tables (DHT), and their

* This work was partially supported under NSF grant 0430769 to Morris.

hash key based lookup approach guarantees relatively fast response time for any given query. However, the rigid network structure compromises the flexibility of data distribution, their key partition based network construction requires aggregation in serial order, and most importantly any logical or physical changes at a peer may cause huge index updates at other peers. Systems like Edutella [7] use hybrid architecture. The network depends on many super-nodes to maintain connection indices and query routings come at the cost of reducing peer connection autonomy.

In this paper, we propose an ontology-based peer exchange network (OPEN) a flexible network organization structure that improves query performance while maintaining a high degree of node autonomy. The OPEN network topology models the interpretation of a world or a specific domain, in which peers are interconnected based on their semantic relatedness, for example their concept descriptions or binary relations between concepts. The logical picture of the OPEN network can be divided into two layers: the knowledge tier, which consists of knowledge nodes, reflects a global ontology for a specific domain, and the local semantic tier which consists of clustered peers offering services and resources semantically equivalent under the same local ontology. Each node in the knowledge tier will maintain a corresponding local semantic tier as an ordered search tree, and peers are ranked using an algorithm based on peer attributes and capabilities, such as resource availability, peer reputation, quality of service, etc. Every peer in the network can decide on its own what data or services to provide to the others, and can decide when to change those services by simply changing its service descriptions. The connection to other nodes is only decided by the knowledge in the global ontology and service descriptions for every peer. Peers need not index the contents of others, so that in a rapidly varying distributed peer exchange network service changes and content updates will not affect the rest of the network and generate a lot of overhead, as happens in many content indexing supernode networks [17].

For Peer to Peer networks, Query capabilities are the main functionality to satisfy various kinds of needs from different members. Most Distributed Hashing Table (DHT) [5, 6] based networks only support exact keyword match and simple range queries. Due to the monotonic nature of hashing functions and diverse nature of digital objects, semantically close related items maybe mapped far from each other in a given key space. However, in practice a peer is more likely to ask for data similar to its previous queries or related to what it provides itself, and the semantic locality layer of the OPEN network will make services to these query patterns more efficient. Based on the rich semantics we use to build the knowledge layer for the network, our queries are not limited to exact keyword matches, or simple range queries. Queries involving relations like typical RDF triple queries are also supported. In many cases, a peer often doesn't know exactly what keyword to use, or what exactly to ask at the beginning of a session, so it might ask general questions like "give me any butterfly images in JPEG2000 format". Based on the result from the network, a human or software agent can learn to make more specific queries. On the other hand, there are synonyms

and different naming systems for similar set of objects, and a restricted keyword match may give no response. It is better to relax the query in a more general form in order to get answers successfully. In any case, a series of related queries will usually be made, and we believe our overlay network structure will route such queries more efficiently than normal DHT networks.

The rest of this paper is organized as follows. Background and related works are provided in section 2. In section 3, we describe OPEN the network structures and construction process. In section 4 we define the query format and routing mechanism. Predicted properties of OPEN network are evaluated by simulation results in section 5. Finally we conclude this paper and suggest future works in section 6.

2 Preliminaries

In this section, we provide basics about P2P characteristics, and review some related studies about semantic overlay networks.

2.1 P2P Characteristics

Milojicic et al. [8] and Risson et al. [9] have pointed out the main features of a peer-to-peer network: decentralization, scalability, self-organization and autonomy. In a P2P network, every peer serves both as a server which offers services to the network and as a client which is able to get information from the network. The peers typically exchange data over some underlying network, e.g. the Internet, and so is an overlay network. The role of each peer is the same. the network is organized and maintained solely by the overlay network model, and no centralized monitoring and control exists which elevate some peers over others. Normally there are huge numbers of query messages, large service consumption and data transfers in the network, and peers joining and leaving. Each peer has full control of what service to expose, what data to share and what connection indices to maintain. Presently, few P2P systems support much influence by a peer over the choice of its peers, and not based on properties of the peer itself and resources it holds. Other challenges that P2P systems need to cope with include anonymity, security and administration transparency are discussed in [10].

2.2 Related Works in Semantic Overlay Networks

In simple terms, an overlay network defines a systematic way to link decentralized peers together and exchange information quickly. In addition to DHT-based overlay network topologies we discussed in the introduction, some efforts have been made to construct semantically related P2P networks. Li et al. [14] have proposed a semantic small world which extends the CAN structure using peer categorizations. In their approach, peers are clustered based on their properties

using a fixed-dimension feature vector and the clustering is done statically. Although similarity between peers can be calculated effectively based on distance in the multi-dimension property space, there are several problems with this approach: resources and data which a peer supplies may not be homogeneous and can not always use a fixed feature vector to categorize them. Moreover, their approach selects the largest cluster as the representative of the peer's character, but this may depend on the semantics of the size metric which may not uniformly categorize the several different kinds of services and data that a peer may supply. Nejdl et al. [7] developed the Edutella system which can respond to queries semantically richer than those supported only by simple keyword match. They use a super-peer overlay structure, and defer the routing and maintenance of the network to all the super-peers. Each super-peer indexes a semantically related key space and all the RDF triples from peers. Basically a query will be forwarded between super-peers to the one containing required keys which will determine which peers hold answers. This architecture is simple and easy to implement, but also has some drawbacks. Super-peers require more resources than their children because they hold all the indices, including those of both the routing and the RDF triples representing the semantic content of the sub-peers. This indexing mechanism is not easy to maintain. Any updates or connection state change of a child peer may lead to substantial index update at a super-peer. In Edutella, nothing enforces semantic consistency between semantic annotations of the same resource by different peers. If a query entails reasoning over the knowledge base represented by the network, this can cause queries to fail even though there are resources which might otherwise meet the query. These disadvantages of Edutella lead us to avoid a super-peer network in favor of one in which all semantic processing is supported locally at the peer. In particular, change at one peer will not dramatically change the interpretation at another peer.

3 Ontology-based Peer Exchange Network

In this section, we describe formal semantics of the OPEN network, and provide technical details and open issues about network construction and maintenance.

3.1 Global Ontology

To enable interoperability between distributed applications which supply heterogeneous data, ontology is widely used to settle the semantic nuances between different parties. Although a peer could choose different ontology meta languages to specify its model, the Resource Description Framework (RDF) [11] and the Web Ontology language (OWL) [12] are widely used to express concepts, knowledge and mappings between ontology elements. In the OPEN network, we use OWL syntax to define the global ontology and peer capabilities.

Definition 1. *A global ontology G is a finite set of shared concepts and relations maintained at bootstrapping nodes for a specific domain.*

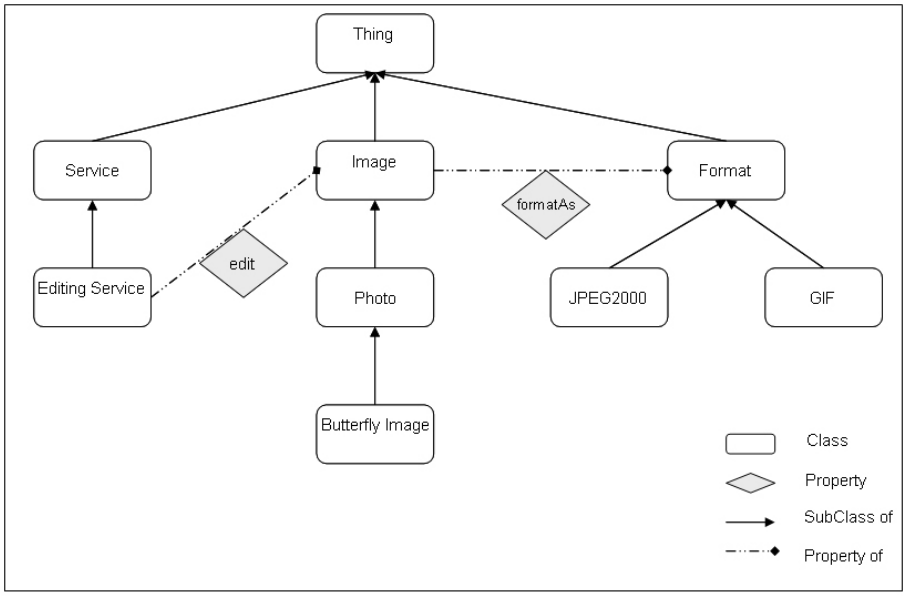

Fig. 1. Sample Global Ontology Structure

Simple sample ontology for collaborating image editing can be defined based on class hierarchies and relations modeled in figure 1 using OWL. This global ontology will be delivered to new joining nodes and help to integrate them to the right place within the network.

3.2 Peer Semantics

Definition 2. *A Peer P in the OPEN network is a triple (D, F, V) where D is an ontological Description of peer's sharing attributes; F is a peer service description which tells other peers how to invoke sharing functionalities provided by P, and V is the set of peer view properties which describe the location of P in the whole network*

Figure 2 illustrates a peer in which D is a description which defines local terminologies extended from global ontology definitions. It describes the objects (butterfly images as defined in the global ontology, offered in JPEG2000 format, also defined there) that peer P will provide to share with others. D also includes equivalence mappings of classes and relations between terminology in P and the global ontology G. The Figure illustrates how D indicates that peer P supports a collaborative image editing service which extends an image editing service from G, and creates a mapping to equate the term "Picture" with the term "Image" in G. D is mainly used while P is entering the network, and will be modified based on its own context changes. F is a set of service interface descriptions.

Ontological Description (D)	<open: Description> <open:Extension> <owl:Class rdf:Id ="CollaborationService" > <rdfs:subClassOf rdf:resource="&globalontology;EditingService" /> </owl:Class> </open:Extension> <open:Data> <rdf:Description rdf:about ="s1"> <rdf:type> <owl:Class rdf:about="&globalontology;ButterflyImage" /></rdf:type> <globalontology:formatAs rdf:rabout ="&globalontology;JPEG2000" /> </rdf:Description> <rdf:Description rdf:about ="s2"> <rdf:type><owl:Class rdf:about="#CollaborationService" /></rdf:type> </rdf:Description></open:Data> <open:Mapping> <owl:Class rdf:Id ="Picture"> <owl:equivalentClass rdf:resource="&globalontology;Image" /> </owl:Class> </open:Mapping> </open:Description
Peer Service Description (F)	<open: ServiceDescription> <open: Service name="s1" type="rdfQueryService" /> <open:Service name="s2" type="agentService" port="7777" /> </open:ServiceDescription
Peer View Properties (V)	<open:ViewProperty> <open:Structurelink> <open:node name="p1" dataid="#s1" knowledgeNode="false"> <open:Link><open:node parernt ="c1" /><open:node child ="p2" /> <open:node child="p3" /> </open:Link> </open:node> <open:node name="c2" dataid="#s2" knowledgeNode ="true"> <open:Link> <open:node child ="p4" /> </open:Link></open:node> </open:node name="r1" dataid="formatAs(#s1,JPEG2000)" knowledgeNode ="true"> <open:Link /></open:node> </open:Structurelink> <open:Semanticlink> <open:Link> <open:node name="r2" dataid="formatAs(XIANG, GIF)" /> <open:Mapping> <owl:Class rdf:id="XIANG"> <owl:equivalentClass rdf:resource="#Photo" /> </owl:Class> </open:Mapping> </open:node></open:link> </openLSemanticlink> </open:ViewProperty>

Fig. 2. Sample Open Peer Profile

V is the peer's view of its interpretation of the network after it has joined. We conceptualize every peer as an agent with different capabilities. F is the place to specify the invocation information for shared data defined in D. In the example, we specify that the ButterflyImage service is a simple RDF based triple query service and the collaboration service is a specific agent service which listens at port 7777. Structural links in V reflect the current logical network connections from P's perspective. For example, P views itself as a node named p1 which has attributes given by the Data object s1 in the description set D, which in turn is described as offering objects of global Class ButterflyImage.

The process of joining the network, described later, results in the node p1 being attached to a number of nodes in a global digraph of concepts and relations, which in Figure 3 we denote as the (global) Knowledge Tier. (The nodes in the Knowledge Tier only come into existence when a peer joins the network). Attachment to a particular point in the Knowledge Layer may be also a property of other peers (for example, all those which offer ButterflyImage objects) and these all form a cluster of all things offering the same object. In turn, those clusters form the Local Semantic Tier. In the view V of Figure 2, node p1 is in such a cluster for the Concept ButterflyImage and is linked to the Knowledge Tier node c1. It has two children p2 and p3 in the Local Semantic Tier Similarly, P also can be represented as p4 for sharing collaboration services and p5 for sharing the FormatAs property with domain butterflyImage and range JPEG2000. In addition to those structural links which reflect logical network topologies, there are semantic links which serve as shortcuts for query routings discovered by past successful query results. Those shortcut links may also include mappings from the peer's ontology to the global ontology. For example, the final node, r2, described in Figure 2 expresses an equivalence of "XIANG" in the node's ontology with "Photo" in the global ontology. (In other words, XIANG is a local synonym for "Photo". Indeed, one meaning of the Chinese word "xiang" is "photograph").

3.3 Network construction

As Figure 3 indicates, an OPEN network consists of two tiers. The first is called the Knowledge Tier, and it maps all the concepts (light gray hexagon) and relations (light gray diamond) from the global ontology G. The black solid arrow denotes subsumption in the concept hierarchy and the black dashed arrow denotes relations between two concepts. The second tier is the Local Semantic Tier, containing clusters of nodes expressing properties of peers, and linked by a knowledge node. Nodes in a single cluster are semantically similar by virtue of being assigned to a single concept from the global ontology, and organized in the form of ordered tree structure.

When a new peer P enters the network, it contacts the maintenance node to get a copy of the knowledge tier connection index. Based on its own ontological description D, it fetches the set of concepts and relations which represents those data or services determinable from the index already represented in the network. For each such concept or relation, P will send a join message to the network. If no node representing that term exists in the knowledge tier, a knowledge tier

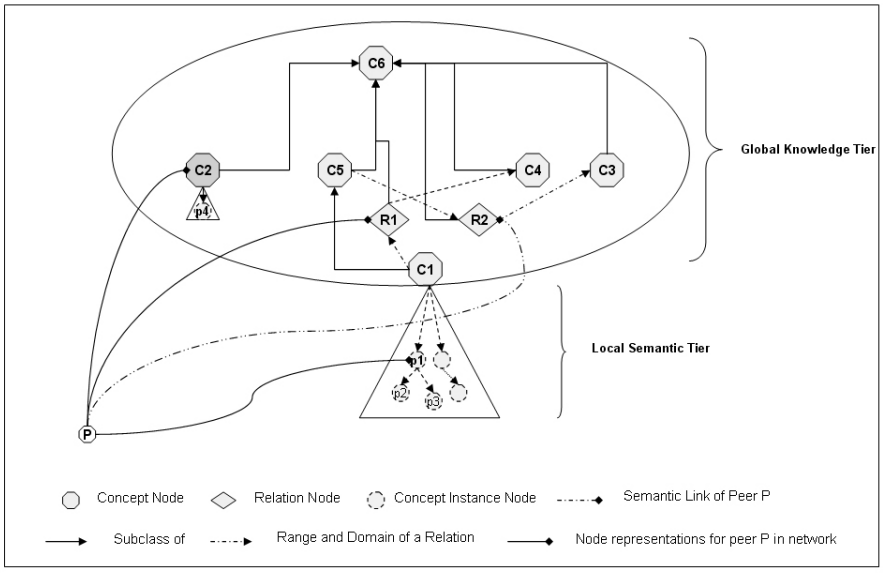

Fig. 3. Open Network Sketch

node will be generated to represent that concept in terms of P (like C2 and R1 in the Figure). However, if an appropriate knowledge tier node does already exist, a node (like p1) will be generated in the semantic tier. Corresponding connection links will be inspected according to definitions in the global ontology and peer' specifications, then added to the peer's View property lists. As we described before, a peer will often supply heterogeneous data and a single cluster is not enough to precisely describe those data. Categorizing data and service descriptions by concepts in the knowledge tier allows a peer to be represented as multiple nodes in the overlay network, each precisely reflecting the exact resources it provides. As we can see, when peer P (shown in Figure 3, described in Figure 2) enters the network, three join messages will be sent and eventually the network contains a knowledge tier concept node C2 which represents collaboration service, a knowledge tier relation node R1 which represents the "formatAs" relation having butterfly images as domain and JPEG2000 format as range, and a cluster layer node p1 which represents butterfly Images. During the network population step, peers will not interfere with each other, hence the construction of the network can be parallelized, and small networks can also be hooked up to make big ones.

3.4 Network Maintenance

There are many considerations regarding network maintenance. As we mentioned, a peer to peer network is a volatile environment. Peers may join and

leave the network arbitrarily and fairly often, and the overlay network structure should adapt to such changes easily. In an OPEN network, nodes may leave the logical network if the description of a peer changes. If a knowledge tier node leaves, it selects a node from its cluster, promotes it to the knowledge tier and provides it with the connection indices to the rest of the cluster. If a cluster node leaves, it chooses a child node and updates indices of both its parent node and the newly selected replacement node. In both cases, the selection criterion is left to the departing node. In our present implementation there is a ranking supplied as children are added to the tree (based on the amount of data they have available), and the decision is always to choose the highest ranking child. Such a peer state change will only cause small connection index changes in a few nodes originally connected to the departing node. This is quite small compared to a supernode leaving a supernode network. As we observed, the content in a P2P network usually changes frequently. Update messages usually cause a lot of system overhead for supernode based networks because any updates at a peer will require corresponding index updates at supernodes. However this is not a problem for the OPEN network since each peer has full control of its own data and generally doesn't index other peer's content. Basically serving a new set of more butterfly images doesn't need to send messages to any others. Moreover, updates about services don't add overhead either and the added service is easy to be located by others. For example, suppose peer P just launches a new collaboration agent listening at port 6666, it only needs to add an entry in its own service description list since there is already a collaboration service in the ontological description, locatable by the knowledge tier concept node for the given type of service. Other peers over the network can just asking the same question "who provides a collaboration service" and are able to locate both agent services hosted at peer P.

In order to make OPEN more resilient to node failure, we use a virtual ring concept for constructing every node in the overlay network. In simple terms, for every node in the Figure 3, we use a ring of multiple nodes each in different peers but represents same concept. By using multiple nodes, we can choose an alternative path to route a query from one knowledge node to another or from a parent cluster node to a child node even if some nodes which deny query forwarding see figure 4.d discussed in Section 5. With this construct, more nodes will be put in every layer of clustering trees, which improves query response time by reducing the tree depth.

4 Query Model

As we mentioned earlier, OPEN supports queries semantically richer than simple keyword matches. For example, a question like "find me all ImageEditing Services" can be represented as $\forall x.\text{ImageEditingService}(x)$ and "find me all butterfly images in JPEG2000 format" as $\forall x.\exists y.\text{ButterflyImage}(x) \wedge \text{FormatAs}(x, y) \wedge \text{JPEG2000}(y)$. For conciseness we use an abstract syntax to describe these queries, but in fact RDF triples form the basis for queries in OPEN. and the abstract

descriptions can be translated to a triple based graph pattern with queries answered by an RDF query language like SPARQL [14]. More complex queries can be composed based on conjunctions. For example, if we want to find a service for collaboratively editing butterfly images in JPEG2000 format, a conjunctive query will be written as:

$\forall x.\exists y, z.\text{CollaborationService}(x) \wedge \text{Edit}(x, y) \wedge \text{ButterflyImage}(y) \wedge \text{formatAs}(y, z) \wedge \text{JPEG2000}(z)$

According to [15], rollup technology can be used to eliminate non-distinguished variables by reducing a tree-like part of a query to a concept without losing semantic consequences and the query can be translated into a SPARQL query.

When a query is submitted to a peer P, it will first check its local semantic links in its peer view property list, V, to find whether there are local indices about the concept or relation in the given query. If such index is found, P will consult the mapping assertions to rewrite the local term to term used in remote peer P' and forward query to P'. If there is no local shortcut, P will consult mapping assertions in the ontological description D,. and rewrite the query by replacing the local term with a global term. It will then forward the query to the corresponding node in the knowledge tier by following the structural links given in V.

Key/value based overlay networks such as those based on DHT require a query agent to have full knowledge about what questions to ask, in this case what the keys are and what they mean. By contrast, an ontologically based network like OPEN permits more generic queries which can be refined by application of the concepts and relations discoverable by the network. By an analogy with human search, the original query can be more generic, and information can be returned to the user to make further queries more specific. Relation inferences like subsumption or instantiation or synonyms can help to formulate more precise questions based on new knowledge acquisition. On the other hand, peers have autonomy to express their models and similar domain knowledge may be specified quite differently at different peers. Mappings between ontologies can help to solve these kinds of problems. However, in an ad hoc network environment, such mappings do not always exist initially. Query relaxation techniques [18] can be used to fetch better results. Our approach to query relaxation is mainly based on the local semantic attributes of peers. For instance, a peer may be knowledgeable about a specific property if it knows a super property. Our current implementation of relaxation relies principally on the use of inheritance through the property and class hierarchies, and the use of wildcards to replace an element in the triples.

As discussed in Section 3, peer local views play an important role in an OPEN network. From a peer's perspective, the links in the peer view V represent local interests of network resources. By caching representations of successful queries in V, OPEN helps to route other similar queries more quickly. At the same time, the typical knowledge tier and clustering structure also make sure that low frequency and not so interesting queries are answered efficiently even absent any caching. (Figure 4.c in Section 5 illustrates that lack of caching typically increases query

depth by at most a factor of 2.) Furthermore, we believe our local routing scheme, which leaves decisions to the peer about what to expose, would have great social acceptance. For example, a series of trust relations could be defined in the peer local view to make sure some queries are only routed to trusted partners. As we saw from earlier discussion, the whole OPEN infrastructure focuses on peer autonomy. However, we can give a peer a slight control over others to improve query performance. Specifically we have implemented a peer ranking mechanism to organize peers in the local semantic cluster. The ranking process can depend on physical connections to take advantage of space locality, logical perspectives about data availability, and annotations about resource-specific confidence in the data, or query feedback. Peers in different parts of the network can use different ranking methods to achieve local optimizations.

5 Evaluation of network

We have done various simulations to verify the predicted properties of the OPEN network. All the experiments are based on a fixed global ontology with 20 classes and 12 relations. Peer profiles are randomly generated based on the global ontology and a fixed set of extensions. All the data services are randomly populated with 100 to 1000 data records. Nodes in a cluster tree are ordered by the number of data records their corresponding peer serves.

Figure 4.a illustrates network load in terms of messages processed for each node in the network, as a function of the ratio of search messages to update messages, for OPEN and two traditional P2P architectures. The result shows that, when most messages are update messages, OPEN has much lower load than a supernode network. As we pointed out above, OPEN allows all the peers to control their own data and is not required to index any other peer‘ content, hence generates less than supernode network message in an update intensive environment. When the message ratio switches to dominance of query messages, a supernode network gets slightly better overall load because full content indices will improve query response time, hence decrease the message loads.

In Figure 4.b, we allow each peer to serve multiple data services, and we observe that the query path length increases with the number of services each peer provides. In Section 3.3, we mentioned that a peer is modeled as multiple mirror nodes in a logical representation of the overlay network. thus, increasing data categories each node provides has the same effect as increasing total number of peers. Moreover, the query path depth is logarithmically related to the total number of nodes in the network (confirmed by the simulation output in Figure 4.b), and query depth scaling is comparable to logarithmic DHT networks like CHORD [6] without restricting peer autonomy as to indexing.

In Figure 4.c, we compare the query performance using local semantic links to not using such links in various scenarios about ratio of search to update messages. In this simulation every peer keeps a semantic record of past successful query results for routing future queries. Experiment shows that using local links will get better performance than only using global index from the knowledge tier

because local semantic links will dramatically improve a series of similar queries by effective shortcuts.

In Figure 4.d, we examine the fault tolerance issue by using virtual nodes. Apparently the result shows that adding the virtual node ring will greatly improve the network resilience by offering alternative paths to forward queries. A virtual node count of only two will greatly improve the fraction of queries that succeed even when half the nodes fail.

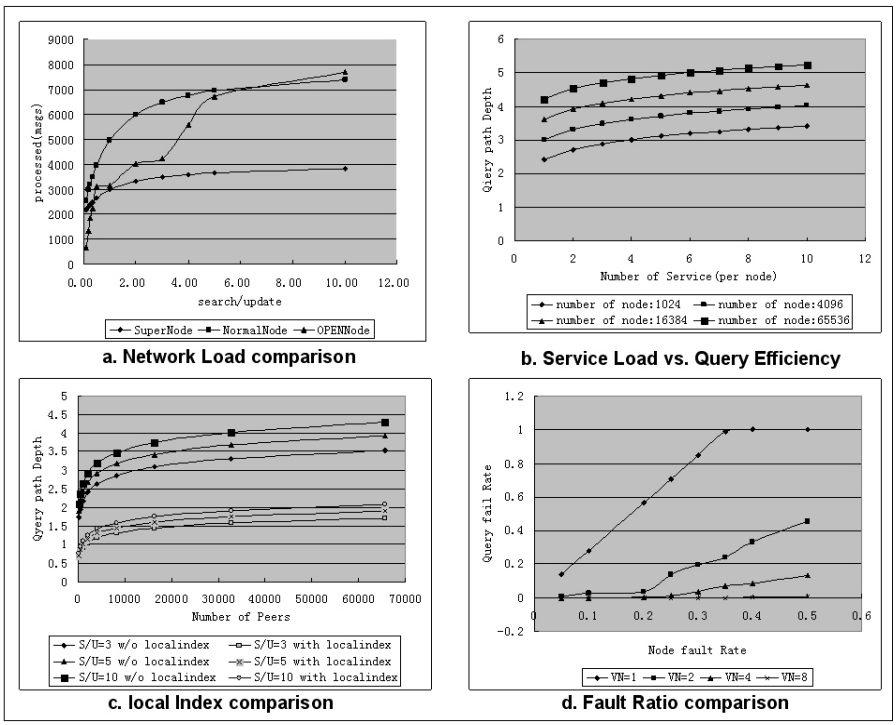

Fig. 4. Network Simulation Result

6 Conclusions and future work

In this paper, we described the architecture and some simulations of OPEN, a new P2P overlay network based on peer ontological descriptions. The network is in use as the messaging infrastructure in a prototype system for collaborative image annotation with sophisticated structured annotation schemas By using an OPEN network, peers can have full control over their behaviors. The simulations

with virtual nodes suggest that high node failure is tolerable, and network maintenance costs, such as periodic heartbeat testing is correspondingly low, and that OPEN networks can be easily cooperate with other distributed systems. Semantically based query mechanisms allow peers to create more complicated models and provide users with polymorphic interpretations of the world. Efficient local query forwarding approaches allows low cost development of special purpose subnetworks.

There are many issues for future work. Instead of caching the whole global knowledge index, which is not efficient if the global ontology is huge, we are exploring holding the necessary information using a hierarchical hashing scheme. Problems dealing with new knowledge population and old knowledge degradation will further affect query performance and network stability. At the cost of adding more complexity to the model, adding hot (i.e. extremely common) complex queries dynamically to the knowledge tier should allow the common queries to be serviced by all the nodes under the corresponding knowledge concept and thereby balance the load among all those nodes that can answer. In such a scenario, the network is adding to its knowledge tier concepts about how people use it. Finally, finding an efficient cluster ranking algorithm may more effectively improve data availability.

References

1. C. Prete, J. McArthur, R. Villars, I. Redmond: Industry developments and models, Disruptive Innovation in Enterprise Computing: storage IDC, February, 2003
2. Napster. http://www.napster.com/
3. Knowbuddy's gnutella faq, http://www.rixsoft.com/Knowbuddy/gnutellafaq.html
4. Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker., Search and replication in unstructured peer-to-peer networks. In Proceedings of ACM International Conference on Supercomputing, pages 84-95, June 2002.
5. S. Ratnasamy, P. Francis, M. Handley, R. M. Karp, and S. Schenker. A scalable content-addressable network. In Proceedings of ACM SIGCOMM, pages 161-172, August 2001.
6. I. Stoica, R. Morris, D. Karger,M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for Internet applications. In Proceedings of ACM SIGCOMM, pages 149-160, August 2001.
7. W. Nejdl, M. Wolpers, W. Siberski, C. Schmitz, M. Schlosser, I. Brunkhorst, and A. Loser., Super-peer-based routing and clustering strategies for rdf-based peer-to-peer networks. In 12th International World Wide Web Conference (WWW'03), Budapest, Hungary, may 2003.
8. D. S. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja,J. Pruyne, B. Richard, S. Rollins, and Z. Xu. Peer-to-Peer Computing. Technical report, HP Laboratories Palo Alto, TR HPL-2002-57, 2002
9. J. Risson and T. Moors. Survey of Research Towards Robust Peer-to-Peer Networks: Search Methods. Technical report, University of New South Wales, UNSW-EE-P2P-1-1, September 2004.
10. N. Daswani, H. Garcia-Molina, and B. Yang. Open Problems in Data-Sharing Peer-to-Peer Systems. In ICDT, 2003.

11. D. Beckett. RDF/XML syntax specification (revised). W3C Working Draft, 2003. Available at http://www.w3.org/TR/2003/WD-rdf-syntaxgrammar-20030123.
12. M. Dean, D. Connolly, F. v. Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness,. Web ontology language (OWL) reference version 1.0. W3CWorking Draft, 2003. Available at http://www.w3.org/TR/2003/WD-owl-ref-20030331.
13. I. Horrocks and P. F. Patel-Schneider. Reducing OWL Entailment to Description Logic Satisfiability. Journal of Web Semantics, 1(4), 2004.
14. E. Prud'hommeaux and A. Seaborne. SPARQL Query Language for RDF, February 2005.W3C Working Draft.
15. U. Hustadt, B. Motik, and U. Sattler. Reasoning for Description Logics around SHIQ in a Resolution Framework. Technical Report 3-8-04/04, FZI, Karlsruhe, Germany, April 2004.
16. R. Morris, M. Calder, H. Dong, J. Huang, Z. Wang, A plugin architecture for distributed collaborative descriptive annotation of biodiversity images Poster accepted for the 5th International Conference on Ecological Informatics, December, 2006, http://www.isei5-conference.elsevier.com
17. B. F. Cooper, H. Garcia-Molina, SIL: Modeling and measuring scalable peer-to-peer search networks, Proceedings of the International Workshop on Databases, Information Systems and Peer-to-Peer computing , 2003
18. C. Tempich, S. Staab, A. Wranik, Remindin': semantic query routing in peer-to-peer networks based on social metaphors, Proceedings of the 13th conference on World Wide Web, 2004