Semantic Overlay Network for Peer-to-Peer Hybrid Information Search and Retrieval

Sungsu Kim
Department of Computer Science and Engineering, Pohang University of Science and Technology (POSTECH), Pohang, Korea
Email: kiss@postech.ac.kr

John Strassner
Division of IT Convergence Engineering, Pohang University of Science and Technology (POSTECH)
Pohang, Korea
Email: johns@postech.ac.kr

James Won-Ki Hong
Division of IT Convergence Engineering, Pohang University of Science and Technology (POSTECH)
Pohang, Korea
Email: jwkhong@postech.ac.kr

Abstract—**Peer-to-peer (P2P) systems have many important advantages. However, most existing P2P systems are limited to providing resource searches based on simple keyword matching, and do not provide any semantic information about the content of the objects stored and the relationships between those objects. This paper proposes the design of a hierarchical semantic overlay network that can be used for content-based full-text search, and is part of our work to use semantics in network management. Our semantic overlay network is based on creating a semantic cluster of objects that is associated with each node in the P2P DHT to provide semantic search. We validate some research questions in our approach by conducting simulations.**

Keywords – Peer-to-Peer System, Information Retrieval, Semantic Overlay Network

I. Introduction

Peer-to-peer (P2P) systems have become popular for the success of applications such as Napster [1], eMule [2], and Gnutella [3]. These applications enable large numbers of users to exchange significant amounts of data. In traditional client-server models, the server holds all the data items, and all clients access the server to get the data items they want to retrieve. However, a node in a P2P system can play the role of both a server as well as a client, which avoids a centralized server bottleneck and a single point of failure. Meanwhile, such decentralized systems have to solve a different set of problems when searching for data in a large network. For example, nodes in P2P systems are continuously joining and leaving, and the system should be able to provide the same services regardless of the current network topology. In order to efficiently maintain such a system, several approaches have been proposed. Gnutella and Random Walk [4] either flood or randomly search the network by using a single path in the network. In terms of the number of messages and the number of hops required to find a node, their approach has high overhead. However, their approach has low maintenance costs, and handling node joining and leaving is relatively easy. Subsequent systems significantly improved search efficiency in P2P systems. Structured overlay networks, such as CAN [5], CHORD [6], and Kademlia [7], use hashed keys for search efficiency. Kademlia is one of the most popular structured P2P networks that use Distributed Hash Table (DHT) systems; for example, Overnet [8], eMule, and aMule [9] all use routing based on the Kademlia DHT, which is called KAD.

All of these approaches address scalability issues with respect to the number of nodes in the P2P system and search efficiency. However, most P2P searching is based on simple keyword matching, which cannot support complex queries such as range queries and queries based on the meaning of a word or phrase. Most systems are limited to finding data objects that include the keywords in their data object name, though some systems [10] get around this limitation by associating an array of attributes with a node. Moreover, queries such as “search for music composed by one artist that sounds like another artist” cannot be supported. The reason is because the routing information does not have any information concerning the content or node semantics. This is because neighbor peer nodes are arbitrarily chosen or assigned based on criteria that do not include semantics.

The primary goal of this paper is to design a semantic overlay network that supports advanced searches, especially those are based on the meaning of a term or phrase. Such a network will have a variety of uses based on the enhanced search and retrieval that it offers. These range from file sharing to social networks to using semantic information to better manage networks. For example, our system will be able to support queries that find similar *behavior* (e.g., find all Denial-of-Service attacks that attacked a set of similar assets, or find all related causes of congestion). Data objects should be represented by a set of attribute values and/or metadata that describe the specific features and/or behavior that is desired to be retrieved. Attributes are decided according to the nature of the applications that are using that P2P system. Based on these attributes, semantically related peer nodes will form a semantic cluster. For example, nodes that have data objects about ‘rock music’ will answer queries related to ‘music’ or ‘rock music’, but not subjects that are not related to rock music, such as ‘diabetes’ or ‘routing’.

There are several challenges in designing our semantic overlay network. First, semantic metrics, not simple keyword matches or even statistical techniques, should be used to construct semantic clusters. However, there is a tradeoff between search speed and complexity. Second, meaning can overlap, which implies that one object can belong to multiple clusters. However, this raises the question of how to form a cluster. Third, searching by keywords as well as by semantic mechanisms should be supported; ideally, the system should be able to determine whether to use a low-cost but simple

978-1-4244-9220-6/11/$26.00 ©2011 IEEE

keyword match or a higher-cost, but potentially more accurate, semantic match.

The remainder of the paper is organized as follows. Section 2 provides background information about P2P systems and semantic clustering. Section 3 gives an overview of the design of a semantic overlay based on the Kademlia DHT, including two important algorithms that are used. Section 4 describes our simulation results to validate our study. Section 5 summarizes the paper and describes future work.

II. Related Work

In this section, we present related work for our semantic overlay network, along with studies about semantic search in P2P networks.

A. Background

In our semantic overlay network, vector space modeling (VSM) [11] and latent semantic indexing (LSI) [12] are used to measure semantic similarity and organize peer nodes into semantic clusters.

1) Vector space modeling (VSM): Documents and queries in VSM are represented as term vectors. Elements of the vector are used to define the importance of a term in the document or query. We used the statistical term frequency inverse document frequency [13] scheme to define the weight of an element. This scheme is presented in Equation 1, where w_{ij} corresponds to importance of j^{th} term in the i^{th} document and tf_{ij} corresponds to the number of occurrences of the j^{th} term in the i^{th} document; N is the number of documents in corpus; df_j is the number of documents that contain the j^{th} term. For example, if the term 'economy' occurred as many or more times in a document than other terms, then the document is likely about economy. However, terms like 'the' cannot be used to decide the characteristics of document because they have no semantic meaning. This is called a "stop word list"; an example of a common one is given in [14]. Therefore, they are first removed before VSM is used.

$$w_{ij} = tf_{ij} \times \log(\frac{N}{df_j}) \qquad (1)$$

Table 1. Example of Terms, Documents, and Query

	Term 1	Term 2	Term 3	Term4
Query	Italian restaurant	menu	N/A	N/A
Document 1	Italian restaurant	menu	Pizza	Pasta
Document 2	N/A	N/A	Pizza	Pasta

2) Latent semantic indexing (LSI): VSM suffers from semantic problems, such as those caused by synonyms and polysemous words. LSI can solve these problems by using statistically derived conceptual indices instead of individual terms. Singular value decomposition (SVD) [14] is used to transform a high-dimensional term-document matrix into a lower-dimensional semantic vector by projecting the former into a semantic subspace. Each element of a semantic vector corresponds to the importance of an abstract concept in the document or query.

Table 1 shows an example of terms, documents, and queries. Keyword matching only succeeds in matching document 1 for a given query. This is because the given query and document 1 contain same terms "Italian restaurant" and "menu". However, document 2 is also an appropriate answer for the query, but is not found because it does not contain either "Italian restaurant" or "menu". LSI provides a measure of the similarity between documents, which is impossible by using keyword matching. In the above example, LSI discovers that document 2 is related with the query using co-occurrence information (i.e., "Italian restaurant", "menu", "Pasta", and "Pizza" co-occur in document 1, which enables us to infer that document 2 is similar to document 1).

B. Structured and Unstructured Overlay Networks

Pure P2P systems define their peers as having identical functionality. Every peer can retrieve files as well as supply them. Examples of this type of system include Gnutella version 0.4, Freenet, and Limewire [16]. The main benefit of this type of system is resilience; any node can join or leave the network without adversely affecting the ability to find content. However, the main drawback is lack of scalability.

Hybrid P2P systems define regular nodes and a set of stand-alone nodes. The regular nodes use metadata to describe their contents, and a central server facilitates the interactions between peers. An example of this type of system is Napster [1]. However, this system has both a single point of failure (the central server) as well as a limit in maintaining the data objects of the peers

Super-peer based systems, such as Edutella [17], contain two types of nodes: super-peers that index content and regular nodes that query for content. A super-peer acts as an indexing server to a set of regular peers (like the hybrid system), called a cluster. Each super-peer indices the content of the peers connected to it. However, super-peers are also connected to each other (as in a pure system), and collaborate by submitting and answering queries on behalf of regular nodes and themselves. This enables a super-peer to forward the query to other super-peers if it cannot satisfy the query. In order to resolve single point failures for super-peers, redundancy is introduced. In this approach, every super-peer that is connected to each other for providing redundancy is also connected to every client, and hence maintains a complete index of all of the client data, as well as the indexes of other partners. The drawback is the additional overhead required for maintaining the super-peer connections, along with the extra traffic generated and processes necessary to keep the indices aligned [23].

Structured overlay networks assign keys to data objects, and peers are organized in a graph to map keys to peers. Efficient discovery of data objects with a given set of keys are enabled by DHT-based P2P systems such as CAN, Chord, and Kademlia. All of these systems provide efficient retrieval of data items based on keywords. Compared to unstructured P2P systems, structured overlay networks are more efficient, because there is no central node to index all the files and no broadcasting of the query to arbitrary nodes. However, these structured P2P systems have not been as widely deployed as unstructured P2P systems. We use a structured overlay network for our semantic overlay network because it supports efficient search, scalability, and can guarantee a key to be found if it exists.

C. *Overlay Network Supporting Semantics*

Some overlay networks, such as pSearch [17] and SSW [19], can support a limited form of semantic search. In pSearch, the semantics of a document are generated by applying LSI on a term-document vector that is generated from a VSM. The CAN P2P network is used for representing their semantic space and indexing. This is one of the first systems that organize content around the semantics of documents contained in the nodes that make up the P2P network. In SSW, peers are clustered according to the semantics of data in each peer, and the clusters are organized as a small world overlay network.

The key difference between our semantic overlay network and these systems is that our system provides better search functionality. Mostly, the dimensionality of VSM used by SSW or pSearch is around 50-300. Therefore, documents that contain rare words are hard to retrieve. However, our system can find a broader range of documents because both keyword matching and complex queries are supported in our system. Moreover, the average path length of SSW and pSearch is longer than that of Kademlia [20]. Although those provide reasonable efficiency, the average path length for search is from 10 to 30. Our semantic overlay is based on a DHT; therefore, keyword matching is provided by the DHT, and more advanced searching is provided by the semantic clusters.

III. SEMANTIC OVERLAY NETWORK BASED ON A DHT

In this section, we describe the concept of a semantic overlay network based on a DHT. In most structured P2P systems, links between nodes are arbitrary. This makes it very difficult to support semantic search. In this paper, we propose semantic clustering of the nodes that make up the DHT.

A. *Overview*

Most P2P systems only support keyword-based searching. Our objective is to define a semantic overlay network that supports keyword-based as well as complex queries. We do this by constructing a hierarchical overlay system – the KAD DHT serves as a logical overlay of the physical nodes, and one or more of the DHT nodes serve as the member of semantic cluster. By building a set of semantic clusters, we form a second set of (semantic) overlay networks, as shown in Figure 1. Each semantic cluster can be thought of as a “community”, which is focused on a particular set of subjects that have a similar meaning.

Given a keyword-based search request, most existing P2P systems find the nodes that have data objects that contain the requested keyword as part of their object name. Simple searches like this are assigned to the DHT level of our hierarchy. Thus, our system will perform as well as Kademlia for these types of queries. However, advanced searches, such as those based on meaning, are instead assigned to our semantic overlay networks. In our semantic overlay network, the contents of a data object will be represented in a VSM format, and the semantic similarity between nodes can be measured by cosine similarity [21]. Nodes that are semantically close to each other are connected

Figure 1 shows semantic overlay network based on the Kademlia DHT. As shown in Figure 1, the underlying overlay network is organized based on the Kademlia DHT, and each node compares its semantic similarity with its neighbors in the DHT routing table; this is called a ‘contact’. Each peer node belongs to a semantic cluster, and peer nodes in the same semantic cluster have similar characteristics. Our semantic overlay network does not explicitly define a semantic cluster, such as by using a tag. However, each peer node implicitly belongs to one or more semantic clusters using the routing information about its semantic neighbors and how similar neighbors are to it.

For example, suppose that peer A in Figure 1 wants to find a document of peer node C by using a complex query. It is impossible to find peer node C using Kademlia contact because the Kademlia protocol only provides mappings between a node and keywords. In our semantic network, peer A belongs to semantic cluster 1, and it can forward the query to peer B, which can forward the query to peer C. Note that peers B and C are in the same semantic cluster 4.

Every peer in this semantic overlay network implements the Kademlia protocol. Peers provide simple keyword matching, and organize Kademlia overlay graphs. Additionally, all peers have semantic vectors, which represent the contents owned by each peer. The semantic vector of a peer is a centroid of data objects owned by the peer. Peers have semantic vectors of other nodes as well. Peers have semantic neighbors on the top of the Kademlia overlay network. Therefore, complex queries, such as “search the music composed by artist A that is similar to artist B”, can be delivered to only those nodes in the set of semantic clusters that are semantically close to the meaning of the query. The details of the overlay construction and data object search will be presented in next subsection.

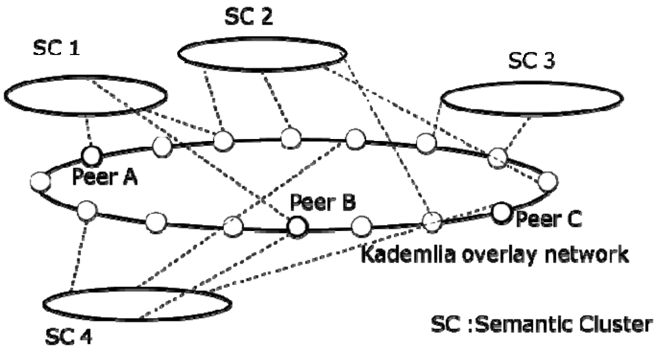

Figure 1. Concept of Semantic Overlay Network

There are several critical issues in designing the semantic overlay network: 1) Semantic clustering – what is the maximum semantic distance allowed for two nodes to be clustered together; 2) Semantic distance algorithm selection – what is the best measure of semantic distance, in terms of accuracy vs. efficiency, for content; 3) Maintenance overhead – how can the overhead incurred in maintaining a semantic cluster be minimized.

In the next section, we will discuss our strategies for creating and managing semantic clusters and for searching data objects.

B. *Semantic Routing*

As shown in Figure 2, the Kademlia routing table consists of a list for each bit of the node ID. If a node ID consists of 128 bits, then the routing table consists of 128 lists. Entries in the list contain the necessary data to locate another node. The data in each list entry is a tuple, consisting of {'IP address', 'port number', and 'node ID'}. Every entry in the list corresponds to a specific distance from a given peer node. Each entry in the list is kept sorted by time last seen. These entries are called k-buckets. The value k defines the upper bound to which every list can grow.

Index	Distance	Contact address
0	$[2^0, 2^1)$	(IP address, UDP port, Node ID)$_{0\text{-}1}$ … (IP address, UDP port, Node ID)$_{0\text{-}k}$
1	$[2^1, 2^2)$	(IP address, UDP port, Node ID)$_{1\text{-}1}$ … (IP address, UDP port, Node ID)$_{1\text{-}k}$
…	…	…
i	$[2^i, 2^{i+1})$	(IP address, UDP port, Node ID)$_{i\text{-}1}$ … (IP address, UDP port, Node ID)$_{i\text{-}k}$

Figure 2. Routing Table of Kademlia Peer Node

In a semantic overlay network, each node stores one or more documents. Searching data objects in a P2P system is equivalent to finding the address of a node that has a certain data object. For simple keyword-based searches, the system must find the ID of the node containing the keyword. However, for semantic searches, the node ID is meaningless, since the search is about the *meaning* of the content, as opposed to matching a word or set of words. Conceptually, our semantic overlay network matches the meaning of the query to the meaning of content, and forwards the query to other nodes that are semantically related to the query. For example, suppose that peer node A has documents about network management, peer node B has documents about Korean history, and peer node C has documents about both network management and Korean history. Our system will forward a query about 'routing' to nodes A and C.

Both keyword-based and semantic searches need to process potentially huge data sets. Recall that each document is described by an attribute vector. This provides two different approaches: (1) search on individual documents, or (2) search on an averaged content of documents. The former is more accurate, but also requires much greater computational and memory resources. The latter provides much better scalability, but its accuracy depends on the homogeneity of the content of the documents contained in a node. Further examination of this fundamental tradeoff is part of our future work.

For semantic labeling, each node obtains the VSM of its own data objects. The calculation of a VSM on each node is done before or when a peer node joins the network. Each node has a *t* × *d* term-document matrix represented as a VSM. Let *d* denote the number of documents, and *t* denote the number of terms in the documents. The semantic distance between documents is measured using cosine similarity. Cosine similarity is a common measure of similarity between a document vector and a query vector. For the semantic distance measurement, a semantic vector normalizes term vectors *X* to unit length ($|X|=1$) in order to compensate for the difference in the size of the documents. The similarity between term vectors *X*=(*x1*,*x2*, ..., *xl*) and Y=(*y1, y2,...,yl*) is defined in Equation 2. Cos(*X, Y*) denotes the cosine of the angle between *X* and *Y*, assuming that *X* and *Y* are already normalized.

$$\cos(X,Y) = \frac{X \otimes Y}{|X| \bullet |Y|} = \sum_{i=1}^{l} x_i y_i \qquad (2)$$

Based on a cosine similarity measurement, each peer node organizes its semantic neighbors and sorts them by similarity. The semantic neighbors of a node are defined using a semantic routing table. Kademlia contacts in Figure 2 are used to find semantic neighbors. Each peer node compares their own contents with their neighbors in the Kademlia routing table (i.e., contact address) using a semantic vector and cosine similarity. Figure 3 shows a semantic routing table, which has semantic vectors and semantic similarities of semantic neighbors in addition to Kademlia routing data. The semantic neighbor list is sorted according to semantic distance between peer and its semantic neighbor in order to access peers which have similar contents.

Index	Distance	Contact address
0	Semantic distance	(IP address, UDP port, Node ID, Semantic Vector)
1	Semantic distance	(IP address, UDP port, Node ID, Semantic Vector)
…	…	…
I	Semantic distance	(IP address, UDP port, Node ID, Semantic Vector)

Figure 3. Semantic Routing Table

In order to complete a semantic routing table, we need a mechanism to determine the semantic vectors of the neighbors of a node and organize them into a semantic cluster. The algorithm for getting semantic neighbors is illustrated in Algorithm 1. When peer node *i* joins a semantic overlay network, peer node *i* first computes its semantic vector matrix V_i. Peer node *i* computes a semantic contact list of Kademlia neighbors using a bootstrap peer. Because our overlay network consists of separate Kademlia and semantic layers, Kademlia routing information should be exchanged first. Peer node *i* requests the semantic neighbors and semantic vectors of peer node *j*, which is peer node *i*'s Kademlia contact. Peer node *i* then computes the semantic distance between semantic neighbors of peer node *j* and itself. If the semantic neighbor of peer node *j* is not in peer node *i*'s semantic routing table, the node address of peer node *j* and its semantic vector are inserted. Entries in the semantic routing table are sorted by semantic distance to peer node *i* in order to organize semantic clusters among similar peer nodes.

Algorithm 1 Peer Joining

```
Peer i joining in semantic overlay network
Extract local semantic vector matrix $V_i$
Get Kademlia contact list from bootstrap peer
for peer j in set of Kademlia neighbors do
    Get Kademlia neighbor j's semantic neighbors and
    semantic vectors
    for peer m in the set of semantic neighbors of peer j
        if semantic neighbor is in semantic routing table of
        peer i
        then put semantic neighbor into semantic routing table
        of i
        else
            drop the neighbor contact
        end if
    end for
end for
```

When a complex query is sent to the semantic overlay network, semantic searching will find documents that are related to the query by their underlying meaning. For example, suppose that the query "architecture for using metadata to manage ubiquitous communications and services" is initiated. Most existing P2P systems only return documents that have the exact same title. That is, if one of words in the query is misspelled, the document will not be found. However, semantic searching enables users to find relevant documents based on meaning, such as by using synonyms or functional relationships (e.g., metadata is, literally, data about data, so if a document uses ontologies in a similar way, that document will be matched, even though the term "metadata" is not present). The algorithm for semantic searching is illustrated in Algorithm 2. When a complex query *Q* initiated from peer *i,* peer *i* computes the semantic similarity between query *Q* and its semantic neighbors. Then, peer *i* selects *K* neighbors that are most similar in meaning to the query *Q,* and forwards the query. When peer *s* receives the query, it computes the semantic similarity between the query *Q* and the documents that it contains. If the semantic similarity exceeds a threshold *t*, the document information will be sent to peer *i*, the originator of the query *Q*.

Algorithm 2 Semantic Searching (complex query)

```
Semantic searching of query Q (K is the default value for
choosing similar neighbors list, t is the threshold for document
similarity with the query)
Complex query Q is received by peer i
Peer i calculates the semantic similarity between query Q
    and its semantic neighbors j
Peer i chooses K semantic neighbors that have the shortest
    semantic distance to query Q
for node s in the set of K semantic neighbors nodes do
    Send query Q to node s
    if node s found documents whose semantic similarity is
    exceeds threshold t
    then return document ID to peer i
    end if
end for
```

IV. Evaluation

This section evaluates the benefits of our semantic overlay. The goal of a semantic search is to find a set of data objects that are semantically similar to the specified query. We built our semantic overlay network prototype to validate our algorithms using the PeerSim P2P simulator [20]. We modified their Kademlia module to implement our semantic routing algorithm. We validated the correctness of our algorithm and overhead that it induced to maintain our system.

A. Simulation Setup

We implemented our semantic overlay network in PeerSim [20], and conducted the simulation experiments on the MS Windows platform. The simulation is initialized by generating a randomly interconnected topology and assigning 160-bit unique IDs to each node, which is the type of ID that is typically used in Kademlia. The semantic space of the LSI implementation typically ranges from 50-130, and we set the dimensionality of the semantic vector to 50. Then, semantic vectors of each node are assigned, and each dimension of the semantic vector is randomly assigned with ranges from 0-1. Finally, we perform a bootstrap process to fill the routing table of each node. For each node, 50 neighbor nodes are randomly selected, and their ID and semantic vector are inserted into each k-bucket. Simulation parameters are shown in table 2. The simulation has been done varying the network size from 200 to 12800 nodes, doubling the number of nodes at each round; each round lasts about 3600 seconds.

Table 2. Simulation Parameter Description

	Descriptions	Values
N	Number of nodes in the network	200-12800
S	Number of semantic contacts in node	20
p	Probability that one new node joins or an existing node leaves for every simulation step (1/100sec)	50%
t	Threshold for query	0.5-0.8
k	Dimension of VSM space	50
K	Query is sent to K semantically closest nodes	3-5

B. Simulation Results

In terms of scalability, we performed a simulation with 200-12800 nodes to show the efficiency of our semantic overlay network. We assume that the underlying network is a reliable transport network. Every second, a number of nodes are randomly chosen and a complex query is generated. We chose this rather aggressive value to stress our design. When a node receives a complex query, it forwards the query to *K* semantically close nodes, and answers the query if the semantic similarity between the query and its semantic vector exceeds a programmable threshold. The cost to compute an answer to the query is measured in terms of overlay path length and latency.

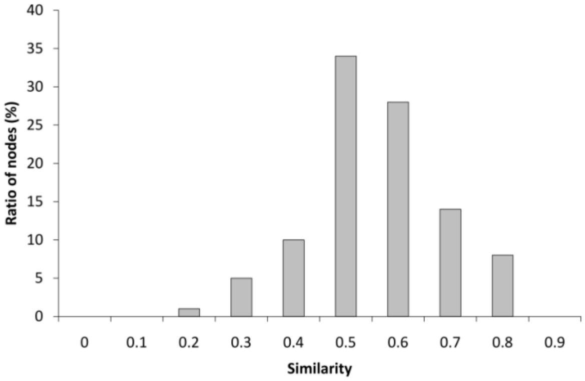

Figure 4. Similarity Distribution of Nodes

First, we show the results of a random query sent to the Kademlia DHT. Figure 4 shows the distribution of semantic similarity values between all nodes in the DHT; this is typical for the semantic distribution of nodes. As shown in Figure 4, most nodes have a semantic similarity in the range of 0.5-0.7, and 34% of the nodes have a semantic similarity between 0.5-0.6. No nodes have a semantic similarity in the range of 0.9-1.0. Therefore, the threshold of the semantic search should be lower than 0.9. We used a threshold value between 0.5 and 0.8. We present the efficiency of our semantic searching algorithm in terms of overlay hop count and latency.

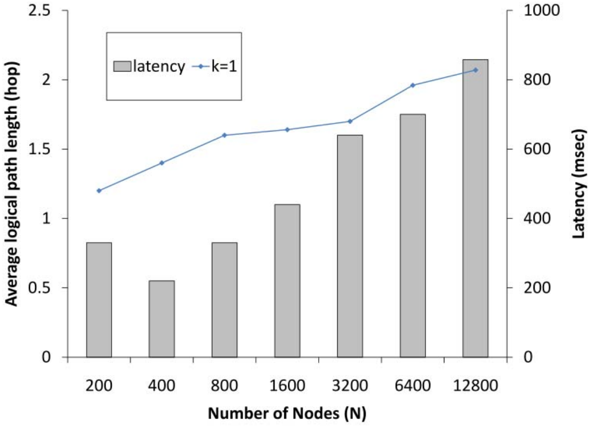

Figure 5. Average Path Length and Latency for Finding Node (*t*=0.8)

Figure 5 shows average path length and latency to reach a destination node with *K*=1. For every search, each node finds the semantically closest nodes and forwards the query. As shown in Figure 4, only 10% of the nodes have a semantic similarity exceeding 0.8. Most of the time, the average path length does not exceed 2. The semantic neighbors of our semantic overlay network are the same as the Kademlia neighbors, and we set the default semantic routing table size to 50. The latency is between 200ms and 800ms, and proportional to path length. The underlying transport layer affects the latency of finding nodes, because the network (which can consist of a large number of nodes) possibly has more physical hops between different pairs of nodes. The complexity of finding a node that has a semantic similarity exceeding 0.8 linearly increases with the number of nodes in the network. The average path length for finding nodes with different thresholds is also tested in the simulation.

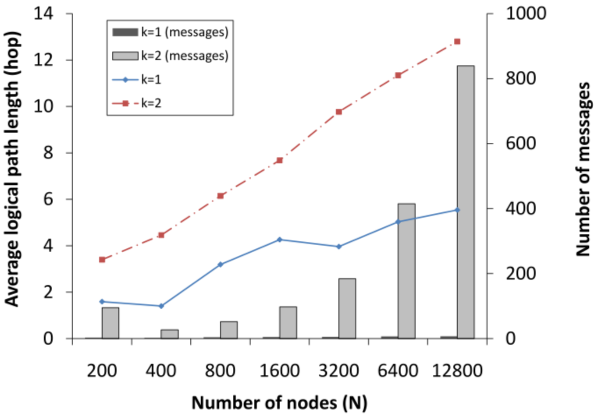

Figure 6. Average Path Length and the Number of Messages (*t*=0.85)

Figure 6 shows that the average path length to find a node whose semantic similarity is 0.85 linearly increases as the number of nodes is increasing. Figure 6 shows the average path length and the number of messages generated for finding a random node, with *K* equal to 1 or 2. In our test set, nodes that have a semantic similarity exceeding 0.85 are 0.01% of all nodes. That is, suppose that there are 800 nodes, then on average, 8 nodes will have a semantic similarity exceeding 0.85. When *K*=1, the average path length and the number of messages generated to find a node is linearly increasing with the number of nodes. The average path length of *K*=2 increases linearly with the number of nodes as well. However, the number of messages generated to find nodes exponentially increases as the number of nodes is growing. This is because the query replicates itself when it passes through the nodes. Theoretically, the number of messages generated is two to the power of the path length. Suppose that the path length is 10 with *K*=2; then, the number of messages is 1024 in the worst case. As *K* increases, the traffic generated by searching dramatically increases, and causes an increase in network overhead. Regardless of the overhead with *K*=2, it returns multiple results because it enables replicating the query.

Figure 7 shows the average path length and the number of results of a single query with *K*=2. The number of results is proportional to the size of the network. The average path length is shorter when a smaller *K* value is used. This is because the average path length is equal to the hop count of a successful query message. For *K*=2, the query is replicated and forwarded to the second most similar nodes, which makes the average path length longer.

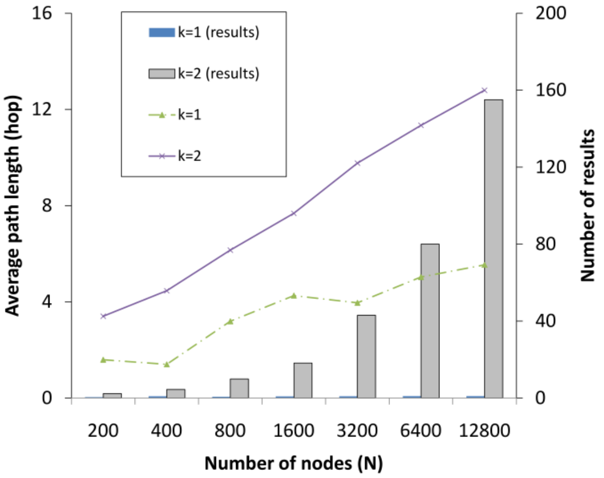

Figure 7. Number of Results Generated and Average Path Length (*t* =0.85)

If each node has a semantic routing table that contains the nodes that have similar content to its own, a query can successfully get the result with *K*=1. The maintenance cost for the semantic routing table is similar to Kademlia's in terms of the number of exchange of messages. In contrast to the Kademlia protocol, which exchanges information consisting of the Kademlia ID, port number, and IP address, a semantic overlay node should exchange semantic vectors, which are much larger than in the preceding case. In terms of the volume of data, our semantic overlay has an overhead comparable to Kademlia.

The simulation quantifies the efficiency and accuracy of our semantic searching. Our major finding is as follows. A semantic search can achieve a reasonable accuracy with good cost with respect to the number of nodes. The semantic similarity is highly dependent on the path length and the number of messages to find a node. *K*=1 guarantees good performance in terms of latency and messaging overhead. *K*=2 gives more results.

V. CONCLUSIONS

This paper proposes the design of a semantic overlay network for content-based full-text search. Two innovative ideas are proposed for our semantic overlay network: 1) semantic clustering; 2) semantic search. each peer in the overlay network are aware of nodes that are semantically close to itself, and forwards queries to nodes that are semantically close to a received query. Our semantic overlay network is designed using LSI and cosine similarity. Our future research will perform a detailed analysis of different semantic metrics, and compare the complexity of the metric to the time and overhead required and also to the accuracy of the results. Scalability will also be tested. Finally, we will investigate how more precise semantics can be used to better classify traffic, and build dedicated semantics service overlays to provide personalized services.

ACKNOWLEDGMENT

This work was partly supported by the IT R&D program of MKE/KEIT [KI003594 , Novel Study on Highly Manageable Network and Service Architecture for New Generation] and WCU (World Class University) program through the National Research Foundation of Korea funded by the Ministry of Education, Science and Technology (Project No. R31-2008-000-10100-0)

REFERENCES

[1] Napster, http://napster.com/.

[2] eMule. http://www.emule-project.net. 2006.

[3] Gnutella, http://gnutella.wego.com/.

[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, Jun. 2002, pp. 84−95.

[5] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network", In ACM SIGCOMM , Aug. 2001, pp. 161-172.

[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, Aug. 2001, pp. 149−160.

[7] P. Maymounkov and D. Mazieres, "Kademlia: A Peer-to-Peer Information System Based on the XOR Metric", 1st International Workshop on Peer-to-Peer Systems, Cambridge, MA, USA, March 7-8, 2002, pp. 53-62.

[8] EDonkey, "Edonkey, overnet homepage", Jan. 2002, http://www.edonkey200.com/.

[9] A-Mule, `http://www.amule.org/`.

[10] A. Crespo and H. Garcia-Molina, "Semantic overlay networks for p2p systems", International Workshop on Agents and Peer-to-Peer Computing (AP2PC' 04), 2004, pp. 1−13.

[11] G. Salton, A. Wong, and C.S. Yang, "A vector space model for automatic indexing", Communications of the ACM, Vol.18 n.11, Nov. 1975, pp.613-620.

[12] R. Price and A. Zukas, "Application of Latent Semantic Indexing to Processing of Noisy Text", Intelligence and Security Informatics, Lecture Notes in Computer Science, Vol. 3495, Springer Publishing, 2005, pp. 602–603.

[13] S. E. Robertson, "Understanding inverse document frequency: On theoretical arguments for IDF", Journal of Documentation, Vol. 50, 2004, pp. 503-520.

[14] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions", Numerische Mathematik 14 (5): 403–420.

[15] http://www.lextek.com/manuals/onix/stopwords1.html.

[16] http://www.gnutellaforums.com/.

[17] Edutella project, http://edutella.jxta.org/.

[18] C. Tang, Z. Xu, and S. Dwarkadas, "Peer-to-peer information retrieval using self-organizing semantic overlay networks", in Proceedings of ACM SIGCOMM, Aug. 2003, pp. 175−186.

[19] M. Li, W.-C. Lee, and A. Sivasubramaniam, "Semantic small world: An overlay network for peer-to-peer search", in International Conference on Internet Protocols, 2004, pp. 228−238.

[20] Stutzbach D and Rejaie R, "Improving lookup performance over a widely-deployed DHT", INFOCOM 2006, 25th IEEE International

Conference on Computer Communications, Barcelona, Spain, Apr. 23-29, 2006.

[21] P. Tan, M. Steinbach, and V. Kumar, “Introduction to Data Mining, Addison-Wesley, 2005.

[22] PeerSim, http://peersim.sourceforge.net/.

[23] J. Strassner, S. Kim, and J. W. Hong, “Semantic Routing for Improved Network Management in the Future Internet”, Recent Trends in Wireless and Mobile Networks (WiMo), Vol. 84, 2010, pp. 163-176.