Content-Based Retrieval in Peer-to-Peer Networks Using Cooperative Caching

Bo Yang
Department of Computer Science
Bowie State University
Bowie, MD 20715 USA
byang@bowiestate.edu

Manohar Mareboyana
Department of Computer Science
Bowie State University
Bowie, MD 20715 USA
manohar@cs.bowiestate.edu

***Abstract*—Devising a uniform paradigm for the representation of multimedia data contents in distributed systems in the presence of heterogeneity, distribution, and semantic gap is a difficult task. When adding technological limitations to this mix, the problem becomes more complex. In this paper, we present a logic-based model for representing the semantics of complex multimedia data objects. The model employs first-order logic to describe the semantic contents of multimedia data, such as visual objects and color/texture features. The aim of this model is to provide general multimedia content representation that can be used in object-oriented information systems. The novelty of this framework comes from (1) its mathematical capability to represent semantic contents, (2) the hierarchical organization and classification of multimedia data objects according to their semantic contents, and (3) the ease of nearest-neighbor searching through synonym links. Finally, the simulation results are presented and analyzed based on various performance metrics.**

Keywords- collaborative computing, multimedia retrieval, logic-based representation

I. Introduction

Peer-to-Peer networks are becoming popular in situations where the infrastructures are either destroyed or too expensive to be built. In a P2P network, each node behaves as a router, forwarding messages for other nodes. The previous researches in P2P networks mainly focuses on designing routing protocols that adapt to the dynamically changing network topology [1-2], and relatively few works have been reported on the data accessing issue [3]. Although the study of routing protocols is important for successful network communications, data accessing is an equally significant issue in the applications of P2P networks, since the ultimate task of a network is to support timely and reliable information retrieval and data sharing among data sources.

One important data accessing application for P2P networks is the *content-based image retrieval* (CBIR). The recent technical advances enable mobile devices to capture and store images, which provide the foundation for image retrieval in P2P networks. The capability of accessing images can drastically enrich the communications between mobile users, improving the quality of P2P data services. However, efficient retrieval of image data in P2P networks is challenging due to the multiple constraints such as node mobility, computation capability, memory space, and bandwidth. Generally, the impact of P2P networks on image retrieval can be categorized as follows:

1) In a P2P network, the nodes communicate with each other in a hop-by-hop fashion; however, the paths between these nodes are constantly changing due to node mobility. An image query may require traversing of the whole network, because the data source nodes are unknown at the requesting node. However, this flooding policy drastically consumes system resources — memory space, network bandwidth, and battery power. Considering the sheer size of image data, the performance of traditional flooding-based policy is even more deteriorated.

2) Scalability and robustness may vary due to different network configurations. A practical P2P network may consist of several data server nodes (data centers) and a collection of client nodes that request data from the data centers [3]. However, this network configuration is not robust or scalable since the data centers behave as hotspots and their movements within the area could also increase the network traffic.

In response to the query, nodes in a P2P network can be partitioned into two groups: the nodes containing relevant data (relevant nodes) and the nodes that do not contain relevant data (irrelevant nodes). In the flooding-based information processing approaches, the query is communicated to all nodes in the system. Alternatively, to improve the performance, one should attempt communication with relevant nodes. This strategy reduces the network traffic and consequently, improves system performance. This paper is intended to address a scheme that limits the communication to the relevant nodes during the course of query processing in a P2P network.

An adaptive semantic-based caching strategy is introduced that keeps track of the recently issued queries and their resolutions. The proposed caching scheme — *Semantic-based Ad hoc Image Caching* (SAIC) — is used to facilitate the resolution of semantically generated queries without unnecessary network traffic. Simulation results show that the proposed scheme can significantly reduce the search cost in terms of query delay and message complexity.

The remaining part of this paper is organized into four sections: Section 2 introduces the background knowledge and related work. Section 3 outlines the preliminary concepts of the caching scheme. Section 4 evaluates the proposed scheme using experimental analysis. Section 5 draws the paper into conclusions.

II. Background

2.1 Image content representation

The representation of image content has been a fundamental problem in image retrieval systems. Considerable research work has been done on extracting and manipulating content information of image data, e.g. image segmentation, classification, and object recognition, to name a few. Most traditional feature-based image retrieval systems employ three types of features in image representation: color, shape, and texture. Color features are widely used in CBIR systems for its simplicity and effectiveness. Typical color features include color histogram [7] and color moment [6]. Shape features are employed in distinguishing images when the contour lines evidently profile the visual objects [8]. Texture features are provided as an important tool for image retrieval. A variety of texture analysis methods have been studied in the past years, such as Daubechies wavelet [9].

However, the performance of feature-based systems is far from satisfactory due to the fact that images with similar features may not share common semantic contents, which is known as the *semantic gap* [10].

To bridge or narrow the semantic gap, one approach is to devise automatic semantic learning functions that map low-level feature space to high-level semantic space [6]. According to the principle of semantic learning, the methods can be categorized as inductive and transductive ones [6]: 1) The goal of inductive methods is to create a classifier which separates some training images based on semantic contents (e.g. annotations) and generalizes well on images without annotations. The most widely used inductive model is support vector machine (SVM) [11]. 2) On the other hand, transductive methods aim at accurately predicting the semantic relevance of the non-annotated images which are attainable during the training process. Methods belonging to this category include latent semantic analysis (LSA) [12], principal component analysis (PCA) [6], and locality preserving projection (LPP) [12].

Although working with different principles, both inductive and transductive approaches provide methods for constructing and training classifiers that are capable of dividing the images (or semantic space) into partitions with linear boundaries. Each partition of the semantic space corresponds to a category of semantically similar images. Figure 1 illustrates the partitioning of the semantic space.

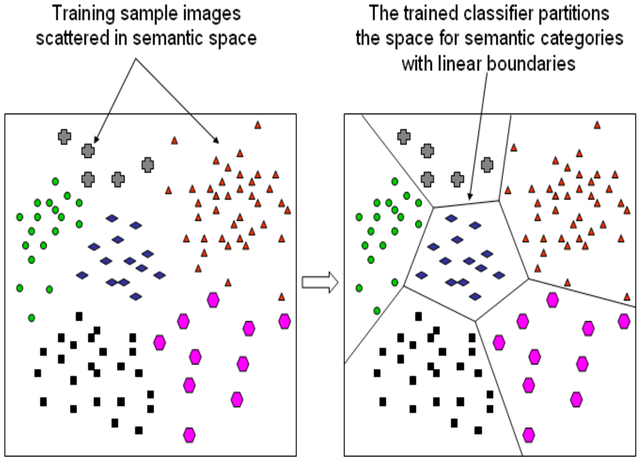
Training sample images
scattered in semantic space
The trained classifier partitions
the space for semantic categories
with linear boundaries

Figure 1. An example of semantic space partitioning.

The partitioning of semantic space provides a means of representing and organizing images based on their semantic contents. Given a collection of semantically similar images, one can collectively represent them using the description of their semantic category. Basing on this observation, we will propose a semantic caching scheme in section 3, with the aim of facilitating CBIR in P2P networks.

2.2 P2P caching

Caching has been widely used in P2P environments to reduce network traffic and deal with disconnections. Although most of the previous study of caching for P2P networks focuses on the efficient exploration of routing information [2], there are a few caching schemes proposed in the literature that facilitate data accessing in P2P networks [3].

This work was supported in part by the Chesapeake Information Based Aeronautics Consortium (CIBAC)

2.2.1 *Data caching*

The data caching approach, which is a natural extension from the caching schemes for wired networks, keeps a copy of the data item that has just been accessed. Traditional schemes let a node cache the results of its recent queries or the data that have been forwarded though it to other nodes [3]. The authors of [13] proposed a semantic caching scheme that allows the caching of queries as the semantic descriptions of the cached data. However, these caching schemes are efficient only for small-size data items, and cannot effectively deal with large-size data such as images in nodes.

2.2.2 *Path caching*

Another approach of caching is to record a path to the data source. This method is usually efficient when the data items are very large and the paths to them are relatively easy to be represented. The authors of [14] examined the allocation of cached data replications in P2P networks. The authors of [3] proposed a CachePath scheme, which dynamically caches the path information of passing-by data. However, the existing path caching schemes consider the data items as independent objects and do not utilize the semantic locality among them. As a result, the content distribution in the P2P network is not fully explored.

Our work differs from the previous related work in that our goal is to devise a caching scheme that facilitates the content-based image retrieval in a dynamic distributed environment such as a P2P network. Hence in this paper, more emphasis is given on efficient locating of data sources that are related with the image query.

III. SEMANTIC-BASED IMAGE CACHING

3.1 *Caching rationality*

The basic idea of the caching scheme proposed in this paper, called Semantic-based Ad-hoc Image Caching (SAIC), is to allow the nodes record concise descriptions of the image query results passing by it. The descriptions, in the form of constraints, characterize the content distribution in the network and reduce the cost of query processing. Figure 2 illustrates the idea of SAIC. Suppose node $\boldsymbol{R}$ issues an image query $\boldsymbol{Q}$ and finds the data source node $\boldsymbol{S}$ through flooding. The query result returned from $\boldsymbol{S}$ will be relayed by a series of nodes (i.e. $\boldsymbol{C}$, $\boldsymbol{B}$, and $\boldsymbol{A}$) to $\boldsymbol{R}$, which forms a chain that divides the network into two partitions. Any later query may go across this chain and meet with one of the relaying nodes. Suppose a node $\boldsymbol{R}^*$ issues a query $\boldsymbol{Q}^*$ semantically similar as $\boldsymbol{Q}$, when $\boldsymbol{Q}^*$ is forwarded to one of the relaying node, say $\boldsymbol{C}$, the data source nodes (i.e. $\boldsymbol{S}$ or $\boldsymbol{R}$) will be determined immediately, and flooding can be avoided. In this sub section, we first define a constraint-based method that summarizes image contents, and then explain the rationality of SAIC in detail.

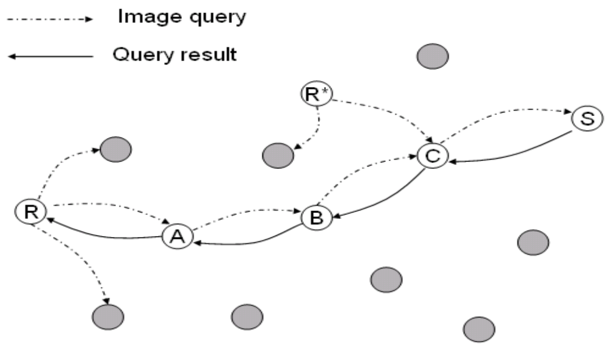

Figure 2. Image query processing and caching.

A. *Semantic image content*

To concisely describe the contents of a collection of images, a representation method is presented within the scope of semantic space. Suppose $\boldsymbol{R}^n$ is the n-dimensional image semantic space. Given a node containing q images $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_q\} \subset \boldsymbol{R}^n$ that belong to r semantic categories $€_1$, $€_2$, …,$€_r$, we give the following definitions for the representation of the images.

Definition 1: Semantic category boundary

Given a semantic category $€_i$, its boundary descriptor $\boldsymbol{B}(€_i)$ can be denoted as a collection of functions showing the polygon boundary of $€_i$, where images within the boundary are assigned 1, and outside images are assigned 0.

$$\boldsymbol{B}(€_i) = \{\boldsymbol{f}_s \mid \boldsymbol{f}_s\text{: } \boldsymbol{R}^n \rightarrow \{1, 0\}\} \quad (1)$$

Besides the Semantic category boundaries, we also give the definition of vicinity constraint to describe image contents more accurately.

Definition 2: Vicinity constraint

Given a set of images $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_q\}$, each image $\boldsymbol{x}_i$ can be represented as a semantic vector $\boldsymbol{v}_i = (a^i_1, \ldots, a^i_n)$. The vicinity constraint $\boldsymbol{C}_v$ is a function that generates a n-dimensional region including all these images:

$$\boldsymbol{C}_v\ (\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_q\}) = ([min(\{a^1_1, \ldots, a^q_1\}), max(\{a^1_1, \ldots, a^q_1\})],\ldots,[min(\{a^1_n, \ldots, a^q_n\}), max(\{a^1_n, \ldots, a^q_n\})]) \quad (2)$$

The semantic category boundaries and the vicinity constraints can be integrated together to form the description of a given set of images. Suppose node $\boldsymbol{N}_i$ contains an image set $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_q\}$, the description can be obtained as follows: First, use the classifiers as mentioned in section 2 to map the images into semantic categories, which are described as $\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots, \boldsymbol{P}_t$. However, the images of a category may not occupy the full semantic subspace of the category. Hence a more concise represented is needed. Let $\boldsymbol{C_{v1}}, \boldsymbol{C_{v2}}, \ldots, \boldsymbol{C_{vt}}$ denote the vicinity constraints of the sub sets of $\boldsymbol{X}$ in the semantic categories, then $\boldsymbol{P_1 \wedge C_{v1}}, \boldsymbol{P_2 \wedge C_{v2}}, \ldots, \boldsymbol{P_t \wedge C_{vt}}$ give more accurate descriptions of each sub set of images, which not only shows their semantic categories but also indicates the variation ranges of the image features. Figure 3 illustrates the representation of images.

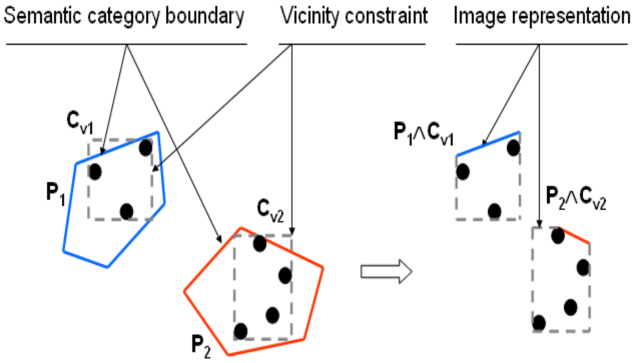

Figure 3. The representation of a collection of images.

B. Cache structure

The aforementioned definitions depict images as data points in the semantic space, which can be collectively described using semantic category boundaries and vicinity constraints. Basing on this representation method, we propose to cache the constraints as the concise description of query results, with the aim of increasing cache hit ratio while decreasing cache space requirement.

Logically, the local cache of a node N_i is divided into a set of cache entries — each entry indicates one or multiple nodes in the network. A cache entry is a tri-tuple (***matching region***, ***vacant region***, ***node list***). The matching region is the constraint-based description of resolved queries, which can be considered as n-dimensional subspaces covering the data points of earlier query results. The vacant region shows the unresolved queries, which can be represented as a collection of subspaces where no query results are found (here we use the Euclidean distance as the semantic distance metric between data points). The node list shows the nodes whose data contents can be characterized by the matching region and the vacant region. Figure 4 illustrates an example of a cache entry.

Physically, we store the constraint-based image content descriptions through paging. The constraints, in the form of polynomial inequations, are stored in one or multiple linked pages. Notice that there are three relationships between the regions described by the constraints: enclosure, overlapping, and isolation. Basing on these relationships, a hierarchical indexing structure can be built on the cache entries, which maintains the semantic descriptions as well as the physical storage information for every cache entry.

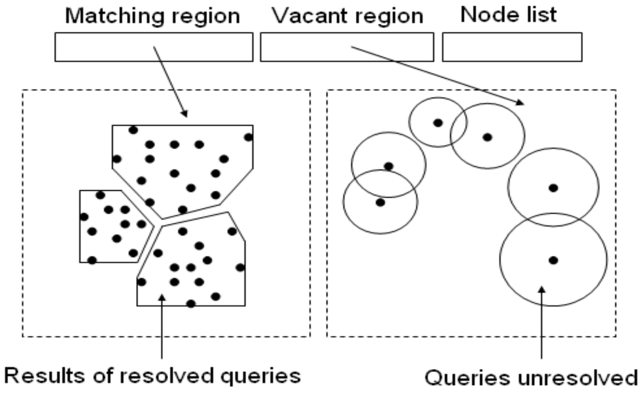

Figure 4. The structure of a cache entry.

3.2 Caching management

As discussed before, image data are cached according to semantic contents, which make the cache management more flexible than traditional schemes. In SAIC, we use a two-phase method to facilitate the effective management of caches, while at the same time avoiding unnecessary network traffic: 1) Initially, the local caches are empty and every query is flooded in the network. The result of the query is forwarded back thru a collection of relaying nodes, where the content description (i.e. the vicinity constraints) is cached for future query processing purpose. 2) The cached description is used to obtain a content distribution overview of the network. When a query is issued to a node N_i, it will first be compared with the matching regions and vacant regions recorded in N_i, and then forwarded to the relevant nodes (i.e. whose matching regions overlap with the query and whose vacant regions do not cover the query). Algorithm 1 shows the details of the cache management.

```
Algorithm 1:   Semantic Cache Management
Input:  a set of nodes $\boldsymbol{S_N} = \{N_1, N_2, \ldots, N_m\}$
        a set of queries $\boldsymbol{S_Q} = \{ Q_1, Q_2, \ldots, Q_s\}$
Initialization: $\forall$ node $N_i \in \boldsymbol{S_N}$, set its cache $\boldsymbol{Ca}(N_i)$ to $\phi$ .
Query processing:
(A) When a query result $\boldsymbol{Re}(Q_j)$ comes from $N_i$:
    if ($Q_j$ is issued by the current node) then
        cache $\boldsymbol{Re}(Q_j)$ and update matching region of $N_i$.
    else compare the requesting node and $N_i$, choose the
        nearer one to the current node to cache
(B) When cache replacement is necessary:
    select two most infrequently visited matching regions,
    merge the matching regions into a larger region,
    compute the intersection of their vacant regions,
    and concatenate their corresponding node lists
```

```
(C) When cache consistency maintenance is necessary:
    if (the data update is an insertion) then
        notify the nodes whose vacant regions overlap the
        inserted image data
    if (the data update is an deletion) then
        notify the nodes whose matching regions overlap the
        deleted image data
```

The main idea of the algorithm is to flood the query when a node does not have proactive knowledge about content distribution, and forward the query only to relevant nodes when enough knowledge is obtained from the cache. Due to the limitation of cache size, the local cache may not have enough space for new query results. Instead of simply dropping the less frequently visited data, we replace them with a coarse semantic description, which represents a larger space that is composed of several smaller subspaces. When data updates occur, we only notify the nodes whose cache validity is affected. Due to the semantic locality, in most cases the insertion/deletion occurs in a small region and the cache validity of other nodes is not affected, hence cache consistency maintenance only adds a trivial load to the network traffic.

IV. Performance Analysis

To evaluate the performance of the proposed SAIC caching scheme, we implemented a simulator in *ns-2* environment (version 2.26) [15]. Since ns-2 does not support content-based retrieval, a semantic-representation module was also developed and added to facilitate image data organization. The simulation results in this paper are based on AODV routing protocol.

4.1 Simulation setup

We used two sets of experimental datasets as the testbeds: a real dataset and a synthetic dataset:

- The real dataset contains 1000 images obtained from various sources, including Corel and Groningen image databases. In the simulator, 50 color histogram features are extracted from each image for the representation purpose.
- To examine the scalability of SAIC in large datasets, we also constructed a synthetic dataset of 16,000 data points whose feature values are assigned by a random number generator abiding by normal distribution in the interval [0, 1].

To allow more flexibility and comprehensive analysis, the simulator relies on a set of input parameters and conditions: The nodes are scattered randomly in an area of 1000*m**1000*m*, moving at speeds randomly selected from $[0, v_{max}]$. The node density can be adjusted by changing the number of nodes in the flat area. The input parameters are summarized in table 1.

TABLE I. The simulation parameters.

Parameter	***Default***	***Range***
Environment size	1000m*1000m	10^4 to 10^8m^2
Transmitter range	100m	100m to 1,000m
Bandwidth	1M bps	
Number of nodes	50	50 to 100
Node mobility (v_{max})	10 m/s	1 to 50 m/s
Local cache size	5 MB	10 KB to 10 MB
Query rate (Q_{rate})	1 query/s	1 to 100 query/s
Feature vector size	1 KB	
Average image size	50 KB	1 to 100 KB
Test dataset size	1000	1 to 16,000
Semantic dimension	50	1 to 100
Nearest neighbors	10	1 to 20

4.2 Simulation results

To evaluate the performance of SAIC, we compared it with two recently proposed P2P caching schemes — CacheData and CachePath [3] — on various metrics.

Cache hit ratio is an important metric for evaluating the performance of caches. Traditional caching schemes rely on large caches and complex replacement policies to achieve high hit ratio. In contrast, SAIC employ vicinity constraints to describe a collection of data objects, which increases hit ratio without large cache size requirement. Figure 5 shows the minimum cache sizes required by CachePath and SAIC to achieve the specified hit ratios. The default parameters are used in this simulation run. In comparison with the CachePath scheme, SAIC has much less requirement on cache size, and does not drastically increase its requirement as the hit ratio increases. The better performance of SAIC in contrast with CachePath stems from its capability of exploiting the semantic locality with vicinity constraints.

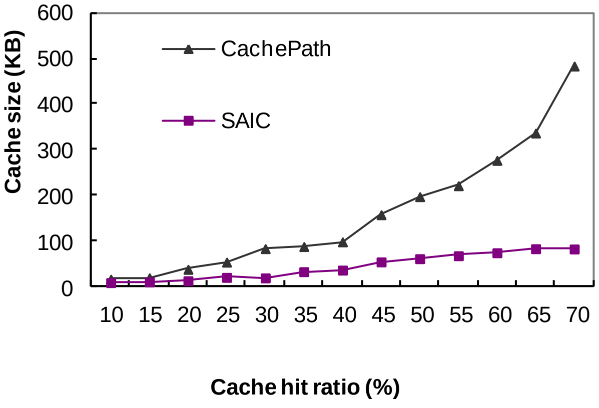

Figure 5. Comparison of cache space requirement.

To analyze the effect of network topology, a series of simulation runs were conducted on various parameters, such as node density and mobility. Figure 6 shows the average query delay as a function of the number of nodes in the network. The delay of all three caching schemes increases as the node density increases, because more nodes compete for limited bandwidth. The paths change more frequently as the node density increases, thus costing the CachePath scheme more time to find the data sources than CacheData. However, with limited cache size, CacheData will incur more frequent cache misses, because it store large-size image data in the cache. SAIC requires much less average delay than both schemes because of its much higher cache hit ratio.

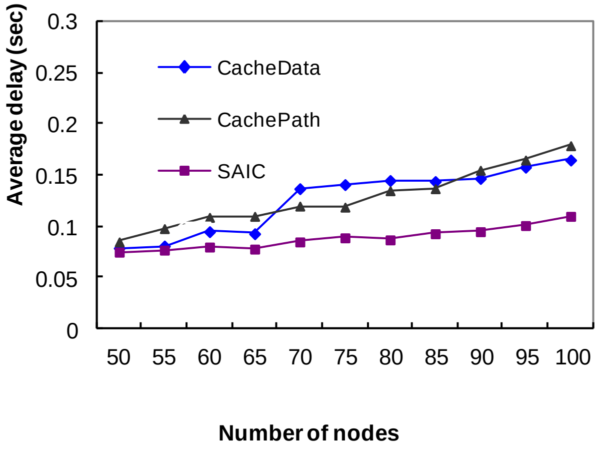

Figure 6. The impact of node density.

The impact of node mobility is shown in figure 7. Similar as the node density, the mobility also incurs more frequent network topology changes. As expected, SAIC resolves queries using relatively much less time than CacheData and CachePath. Notice the average query delay of SAIC only slightly changes as the node mobility increases, which indicates the steady quality of SAIC.

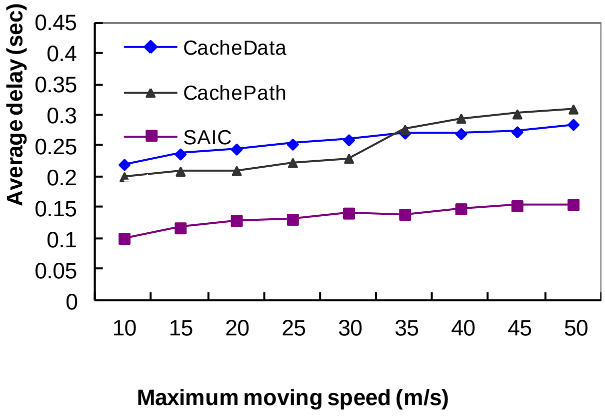

Figure 7. The impact of node mobility.

In a P2P network, the number of messages is often used to evaluate the cost of resolving a query. The messages involved in the query processing can be divided into two categories: the control messages used for transmitting image queries (i.e. feature vectors) and the data messages used for transmitting query results (i.e. image files). The data messages are comparatively much larger than control messages, thus causing more network traffic. The average number of control messages required to resolve a query is shown in figure 8, while figure 9 depicts the average number of data messages per query. As shown in figure 8, CachePath and SAIC require more control messages than CacheData because they need to forward the query to the data source nodes. However, due to the limitation of cache size, CacheData incurs more cache misses, causing much more data messages. In contrast, SAIC incurs the least of data messages because of its capability of forwarding the query only to the nodes with relevant data contents.

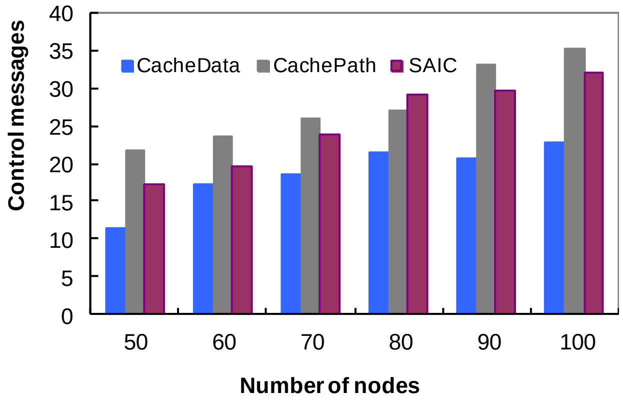

Figure 8. Average control messages per query.

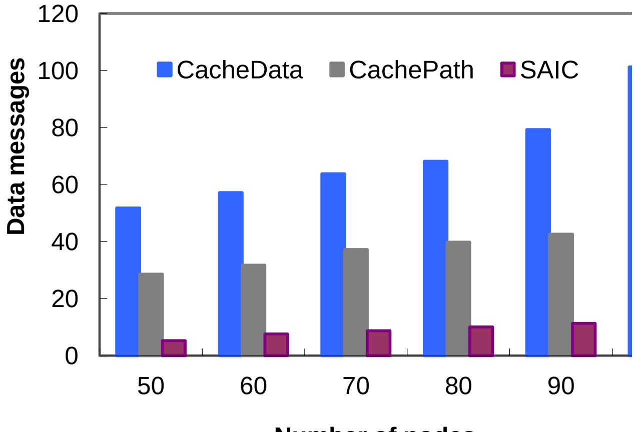

Figure 9. Average data messages per query.

V. CONCLUSIONS

Content-based image retrieval is a challenging task in P2P networks due to the mobility, the bandwidth, and the lack of infrastructure. In this paper we propose a semantic-based caching scheme SAIC to facilitate the efficient image retrieval in P2P networks. It employs vicinity constraints to represent image contents, increasing high cache hit ratio while reducing the cache space requirement.

SAIC has several desirable features: It achieves high cache hit ratio without incurring much network traffic. The average query delay is reduced, which means better image retrieval service quality. The performance of SAIC does not change drastically with different network settings (e.g. node density and mobility), which shows the robustness and scalability of SAIC.

We are tuning the performance of SAIC further and exploiting its application in other P2P environments such as sensor networks. In addition, although SAIC is proposed for the efficient retrieval of images, it can also be extended to accommodate other multimedia data (e.g. audio and video data), considering the similarity of retrieval approaches for different multimedia modalities.

REFERENCES

[1] C. E. Perkins, E. M. Royer, S. R. Das, M. K. Marina. Performance comparison of two on-demand routing protocols for ad hoc networks. IEEE Personal Communications, 2001, 8(1): 16-28.

[2] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. ACM Mobicom, 1998: 85-97.

[3] L. Yin and G. Cao. Supporting cooperative caching in ad hoc networks. IEEE INFOCOM, 2004.

[4] H. Luo, R. Ramjee, P.Sinha, L. Li, and S. Lu. UCAN: A unified cellular and ad hoc network architecture. ACM Mobicom, 2003: 353-367.

[5] V. Dheap, M. Munawar, and S. Ward, Parameterized neighborhood based flooding for ad hoc P2P networks. IEEE Milcom, 2003: 1048-1053

[6] J. He, M. Li, H. Zhang, H. Tong, C. Zhang. Manifold-ranking vased image retrieval, ACM Multimedia, 2004:9-16.

[7] M. Swain and D. Ballard. Color indexing. Int. Journal of Computer Vision, 7(1):11-32, 1991.

[8] C. Schmid. A structured probabilistic model for recognition. Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 1999.

[9] J.-Z. Wang, G. Wiederhold, O. Firschein, and X. W. Sha. Content-based image indexing and searching using Daubechies's wavelet. Int. Journal of Digital Libraries, 1(4):311-328,1998.

[10] G. Auffret, J. Foote, C. Li, B. Shahraray, T. Syeda-Mahmood, and H. Zhang. Multimedia access and retrieval: The state of the art and future directions, ACM Conference on Multimedia, 1999: 443-445.

[11] P. Hong, Q. Tian, and T. Huang. Incorporate support vector machine to content-based image retrieval with relevance feedback. Proc. IEEE Int. Conf. on Image Processing, 2000:750-753.

[12] X. He, O. King, W.-Y. Ma, M.-J. Li, H.-J. Zhang. Learning a locality preserving subspace for visual recognition. Proc. IEEE Conf. on Computer Vision, 2003.

[13] Q. Ren and M.H. Dunham. Using semantic caching to manage location dependent data in mobile computing. ACM Mobicom, 2000: 211-221.

[14] T. Hara. Efficient replica allocation in ad hoc networks for improving data accessibility. IEEE INFOCOM, 2001.

[15] NS notes, http://www.isi.edu/nsnam/ns/. 2004