

# HON-P2P: A Cluster-based Hybrid Overlay Network for Multimedia Object Management

Mouna Kacimi, Kokou Yétongnon, Yinghua Ma, Richard Chbeir

Laboratoire LE2I

University of Bourgogne

Sciences et Techniques

21078 Dijon Cedex France

kacimi@khali.u-bourgogne.fr

{kokou.yetongnon, yinghua.ma, richard.chbeir}@u-bourgogne.fr

**Abstract**—Multimedia centric P2P must take into consideration the main characteristics and the complex relationships among multimedia objects. In this paper, we propose a cluster-based Hybrid Overlay Network HON-P2P for sharing multimedia content. It consists in clustering peers with similar feature based or semantic properties. We define two types of clustering methods corresponding to the semantic and feature based overlays: semantic clustering and feature based clustering. To improve the information retrieval in the HON-P2P network we propose a multimedia cache management methodology. Semantic multimedia cache and feature based multimedia cache are defined for each type of overlay. Moreover, we study the cache placement possibilities inside the cluster and we propose a cache distribution technique taking into account peers capabilities and range query

[10], [14], [20], [23], [39], which in turn send the query to their neighbors, continuing the flooding process until all peers are gradually reached. In the *document routing* model [9], [17], [18], [36], an identifier  $Id_p$  is assigned to each peer using a hash code. Similarly, an identifier  $Id_d$  is assigned to documents using a hash code based on its properties (name, size, keyword, etc.). The similarity between the document and peer codes is calculated and used to route documents and queries to the relevant peers.

In this paper we will focus on cluster based P2P architectures for multimedia data sharing and processing. P2P architectures can be viewed as logical overlay network built on top of the underlying physical links of the Internet network. The early P2P architecture are flat unstructured overlays based on centralized index servers or decentralized flooding query algorithms. Structured overlays are design alternatives that attempt to alleviate some of the problems that plague the early unstructured random overlays, namely the problems of (1) high processing and storage overhead of centralized servers and (2) high network load and delay of random unreliable query search scheme. One approach for creating structured overlay is to build a hierarchy on the underlying unstructured overlay. The result is a multi-layer architecture where peers (super peers) at one level control and manage the peers at the lower level. The hierarchy contains two types of nodes: the super peers with high storage and computing capabilities carry out query routing, indexing and data search on behalf of the less powerful peers. Another approach is to cluster peers based on common interest and characteristics. This solution imposes a graph structure on the underlying overlay. Two nodes or peers are linked in the graph if they share some common characteristics. In section 2 we present different characteristics for creating structured overlays networks: the network parameters, context information, and content related metadata.

## I. INTRODUCTION

In recent years, there has been an ever increasing interest in the development of P2P systems, spurred by the popularity of content sharing applications (e.g. Napster [27], eDonkey [11], Kazaa [20], Gnutella [14]), distributed computing applications (e.g. SETI@home [38], Avaki [7], Entropia [12]), and design platforms used to support the development of P2P networks (JXTA [19], .NET [30]). P2P systems consist of collections of hosts or peers organized in distributed scalable environments which supports the pooling and sharing of resources (processing power, content, storage etc.) and in which peers can autonomously join or leave.

P2P systems have raised several issues and challenges and have been the focus of many research studies. One issue that has received much attention is the specification of P2P models. Dejean Mulojicic et al in [26] classify P2P models in two categories: pure and hybrid. The pure models (e.g. Gnutella [14], Freenet [17]) are decentralized while the hybrid models (e.g. Napster [27], Groove [15] etc.) include some level of centralized information registries which are used to identify peers and locate objects requested by users. Another issue central to P2P systems is information retrieval. Several classes of query routing model have been investigated. The *centralized directory* model records information on both services and contents provided by peers [7], [8], [11], [27], [38]. The basic idea of the *query flooding* model is that a peer sends out a query to all of its directly connected neighbors

We propose a Hybrid Overlay Network P2P architecture (HON-P2P) for sharing multimedia content. Using low level feature-based and semantic characteristics, peers are grouped in clusters based on the similarity of their contents. The architecture allows two types of structured overlays. Semantic overlays are based on ontologies or concept classification hier-

archies used to describe the semantic of peer contents while the feature-based overlays are built on multimedia content features such as color and shape.

The architecture of a cluster is a two level hierarchy consisting of super peers and simple peers. The super peers, which are responsible for the management of the clusters, have high processing and storage capacities while the simple peers have limited capabilities. A peer can join different clusters in different overlays. To improve the performance of information retrieval in the HON-P2P, we propose two types of multimedia cache: a semantic multimedia cache assigned to semantic clusters and a feature-based multimedia cache assigned to feature-based clusters.

We make the following contributions by proposing:

- a hybrid overlay network that takes into account both the semantic and low level features of multimedia systems. An overlay is composed of one or more clusters of similar peers.
- two methods for summarizing the low level representation of peers and clustering peers according to the similarity of content. The first method defines a partition of the feature space into cells and use the distribution of documents over the partition cells as the basis for defining peer similarity, considering two peers as similar if their contents are distributed on the same sub regions of the feature space. The second method uses the relative weight or frequency of document occurrences in different partitions of a feature to compute peer similarity.
- a semantic clustering method based on classification hierarchy (taxonomy). The content of each peer is described by a weighted vector corresponding to the elements of the concept taxonomy.
- a multimedia cache management methodology that shares query traces among peers to improve the information retrieval. We have defined two types of multimedia cache for each type of overlay: semantic multimedia cache and feature-based multimedia cache.

The remainder of the paper is organized as follows. In the next section, we discuss different peer clustering approaches. Section 3 presents the hybrid overlay network for managing multimedia data in P2P environment. Section presents the HON-P2P overlays and the semantic and low level feature clustering techniques. In section 5, we discuss cache management issues. And finally section 6 concludes the paper.

## II. BACKGROUND

Clustering involves the creation of links on top of unstructured P2P overlay networks to group peers with similar content. Figure 1 presents different categories of clustering approaches.

### A. Context-based clustering approaches

Different context-based (network, peer and application) information or properties can be used to create clusters of peers. For example, some clustering approaches are based on network related information. In [21], Krishnamurthy et al define a cluster-based P2P architecture called CAP by assigning peers which share IP address prefixes to the same cluster. Similarly,

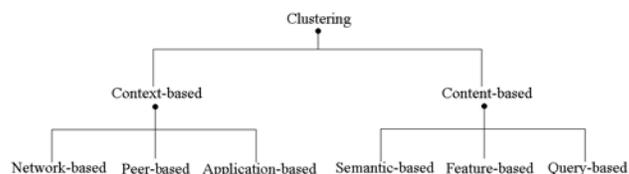


Fig. 1. Categories of clustering techniques

Bestavros et al [1] use DNS (Domain Name Server) information to group web clients served by the same DNS. Their clustering method requires web server usage log or DNS usage pattern. Other approaches use peer characteristics instead of peer content descriptions [5], [19], [24]. For instance, the open JXTA Search framework [37] allows clustering by both geographic and property similarities. Agrawal et al [5] use a distance metric based on network latency for clustering. A cluster consists of a set of peers such that the distance between any pair is less than a given diameter  $D$ . Loser et al [24] use static or dynamic peer properties. The static properties can be a query or result schema or IP domain shared by the member peers while the dynamic properties can be the types and number of resources of a peer. Another method for creating clusters is to consider the functionalities or properties of applications. For example, Wang et al in [42] propose a P2P architecture called Friends Troubleshooting P2P Network (FTN) for resolving machine configuration problems. When a peer or machine encounters a configuration problem, it can request help from a cluster of trusted friend peers to obtain correct versions of configurations. The friend network is based on sharing the same family of configurations. Similarly, Marti et al [25] propose a friend network of peers for dealing with security related attacks. The friend network can use existing social network services (like AOL, Microsoft, Yahoo) to create trusted social links among peers.

### B. Content-based clustering

Content-based clustering exploits similarities among the documents of peers. They require an appropriate representation of peer content for assigning peers with similar content to the same clusters. Several approaches have been proposed in this direction [4], [16], [22], [28], [29], [31], [41].

- *Semantic-based* approaches [4], [28], [29], [41] associate peers with semantic descriptions that can be simple key word based annotations, schema or ontologies. The descriptions are usually based on common domain concepts. Crespo et al [4] propose Semantic Overlay Networks (SON). It uses classification hierarchies to cluster peers which have semantically similar content. A semantic overlay network is associated with each node in the classification hierarchy. The authors discuss several strategies for assigning peers to SON. The conservative strategy puts a peer in a SON if it has at least one document classified in the corresponding concept while the less conservative strategy places a peer if it has a significant number of documents corresponding to the concept. Nejd et

al address clustering strategies for RDF-based P2P networks in [29]. Their solution is a schema-based approach where the content of peers are annotated using RDF and RDF-schema. The RDF P2P network is a hierarchical architecture consisting of super peers interconnected by a hypercube topology and peers connected to the super peers. When a peer joins the network, it publishes a metadata based description of its content to the super peers. The RDF-based descriptions are used to carry semantic comparison between peer and super peers. The Piazza project [41] is a peer data management system (PDMS) aimed at sharing semantically heterogeneous data and schemas. Clusters are built by creating mappings between semantically similar peers.

- *Feature-based* approaches create clusters based on low level feature descriptions of peer content. The documents and the peers are commonly represented by feature vectors; For example, Hang et al [31], [39] propose a CBIR (Content-Based Image Retrieval) system on top of a P2P network in which peers that share similar images are grouped together. Each peer extracts the content-based of its shared images to form a collection of feature vectors which represent a signature value of a peer. The signature vectors are used to calculate similarity measures between peers. Two types of links can be established between peers. A random link connect a peer  $p$  to another peer in a random manner while an attractive link is an explicit connection a peer makes to another peer which have similar images.

- *Query-based* clustering approaches [22] are based on peer request traces. A peer uses request relationships to other peers to construct semantic links to them. Sripanidkulchai et al [22] propose a technique called shortcuts to implement a performance enhancement layer on top of the flooding based content location mechanism of Gnutella [14]. Each peer creates and maintains its shortcuts list based on its request trace. Shortcuts are ranked utilization metrics such as the probability of providing relevant content, latency of the path to the shortcut, available path bandwidth, shortcut load etc. Handurukande et al [16] investigate real query traces collected in the eDonkey 2000 peer-to-peer network using different strategies that exploit semantic proximity between peers.

### III. THE HON-P2P ARCHITECTURE

We present a hybrid architecture for clustering peers that share similar contents. We take into account the semantic and feature-based characteristics of multimedia objects. Figure 2 depicts the structure of the HON-P2P Hybrid Overlay Network, consisting of one or more overlays. The overlays are shown as boxes and clusters are shown as ovals within the overlays. Each overlay includes one or more cluster of similar peers. We distinguished two types of overlay. The semantic overlays corresponds to domain classification hierarchies which define concepts shared by the peer clusters. Similarly, feature-based overlays are created to represent properties of multimedia data such as color, texture. The component clusters of the overlays are constructed using different clustering methods which we discuss below.

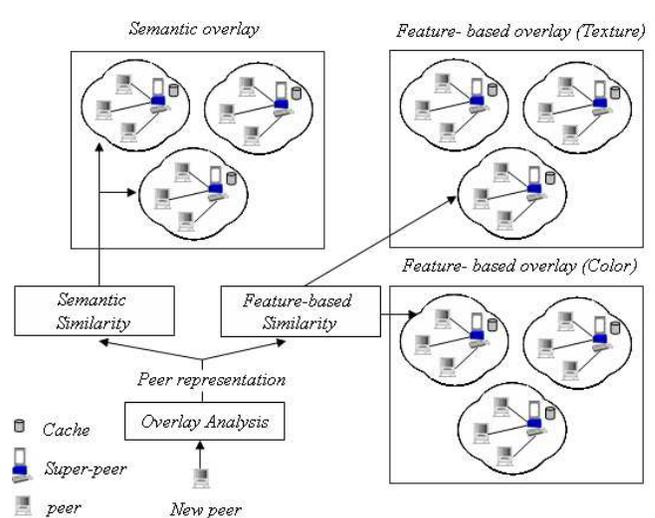


Fig. 2. Hybrid overlay networks and peer clusters

As shown in figure 2, a peer can join more than one cluster in different overlays. To join a cluster, the peer must first carry out an overlay analysis to determine its semantic or feature-based representation or signature. It then connects to any peer in the HON-P2P network to obtain overlay information. Once connected, the peer sends through the initial connection its signature to interested super peers which reply with their cluster signature data. Based on the returned cluster representations, the peer chooses to join and create links to one or more clusters. Finally, the super peers of the clusters joined by the new peer recompute their representation values. When a peer leaves the HON-P2P network, it sends notify using any query processing method (e.g. flooding) its directly connected neighbors and the super peers of its cluster groups.

#### A. Architecture model of a hybrid overlay network

The architecture of an overlay cluster is a two level hierarchy consisting of super peers and simple peers. The super peers, which are responsible for the management of the clusters, have high processing and storage capacities while the simple peers have limited capabilities. A peer can join clusters of different overlays.

##### 1) Description of the components of the overlay network:

The overlays, peers and clusters consist of the following:

- *Peer definition:* a peer is described by  $P (Id, S, \{O\}, \{Id_{SP}\})$ , where  $Id$  is a unique identifier associated with  $P$ , it is used to differentiate a peer from any other peer.  $S$  is a signature value representing the characteristics of a peer. The signature contains a semantic description and/or a feature-based description of the multimedia objects stored in the peer. It consists of network characteristics (the available bandwidth, the rate, etc), hardware and software characteristics and peer load.  $\{O\}$  is the set of Multimedia Objects stored in the peer.
- *Cluster definition:* a cluster is described by  $C (Id, S, Id_O,$

$\{P\}$ ,  $\{Id_{SP}\}$ ,  $\{Id_C\}$ ), where  $Id$  is a unique cluster identifier assigned to  $C$ .  $S$  is a signature value representing the description of the peers belonging to the cluster.  $Id_O$  represents the identifier of the overlay to which the cluster belongs.  $\{P\}$  is the set of peers constituting the cluster.  $\{Id_{SP}\}$  represents the set of peers capable of managing the cluster. Only one peer has an active status and represents the super peer. The others are candidate super peers which can become active when the current super peer is down.  $\{Id_C\}$  is the set of neighbor (most similar, adjacent clusters, etc) clusters. This information is important for routing query between clusters.

- *Overlay definition*: an overlay is described by  $O (Id, S, \{Id_C\})$  where  $Id$  is a unique identifier associated with  $O$ .  $S$  is a signature value describing the type of the overlay (semantic or feature-based) and the representation of clusters containing in the overlay. For example the semantic clusters can be represented by ontologies or concept classification hierarchies while the feature-based clusters can be represented by color features, shape features, etc.  $\{Id_C\}$  is the set of clusters in the overlay.

2) *Functional description of the super peer*: In addition to peer management functions such as maintaining state information of peers, the super peer of a cluster carries out the following functions:

- *Cluster signature computation*: the super peer computes the signature of the cluster based on the signatures of the connected member peers. The cluster signature is updated when a peer join or leave the network.
- *Cache management*: the super peer maintains the cache and is responsible for its operation: query admission, cache replacement, etc. More details of the caching method are given in section 4.
- *Query processing and routing*: when the super peer receives a local query, it analyzes the descriptions of the peers belonging to its cluster and select the most similar to the query. The list of the selected peers is sent to the requesting peer. When the super peer receives an external query, it computes the similarity between the query and the existing clusters to select the relevant peers.

#### IV. HON-P2P OVERLAYS AND CLUSTERING

In this section, we describe two feature-based clustering methods and a semantic clustering method.

##### A. Partitioning the feature space

The goal of the clustering scheme is to group in the same clusters peers whose documents are similarly distributed in a feature space defined by the low level features  $f_1, f_2, \dots, f_k, \dots, f_n$ . The basic idea is to define a partition of the feature space into cells and use the distribution of documents over the cells as the basis for defining peer similarity, creating clusters and computing query similarities to peers and clusters. Thus, two peers are considered similar if their contents are distributed on the same sub regions of the feature space. To define a partition of the feature space into cells, we evenly divide the range of values  $[f_i^{min}, f_i^{max}]$  of a feature  $f_i$  into

$m_i$  intervals of size  $\lceil \frac{f_i^{max} - f_i^{min}}{m_i} \rceil$ , for  $i = 1, 2, \dots, n$ . We denote the resulting set of cells  $\phi = \{\varphi_1, \varphi_2, \dots, \varphi_m\}$ , where  $m$  is given by  $m = \prod_{i=1}^n m_i$ .

**Peer content representation**: Consider a peer  $P_i$  and its document set  $D = \{D_{ij}\} = \{[f_{1j}, f_{2j}, \dots, f_{kj}, \dots, f_{nj}]\}$ , where  $f_{kj}$  is the  $k_{th}$  feature value of  $D_{ij}$ . Each document  $D_{ij}$  is mapped to one cell of the feature space. If we denote  $\alpha_{ik}$  the number of documents of the peer  $P_i$  in the partition cell  $\varphi_k$ ,  $k = 1, 2, \dots, m$ , we can describe the content of  $P_i$  by a signature vector  $S_i$  defined over the feature space cells as  $S_i = [\alpha_{i1}, \alpha_{i2}, \dots, \alpha_{ik}, \dots, \alpha_{im}]$ . The signature vector records the distribution of the content of a peer over the cells. It defines the density of each cell with respect to the peer. This notion of cell density is used for mapping a peer to a cell. Figure 3 shows the partition cells of a 2-dimensional feature space using the low level features  $f_1$  and  $f_2$  and the distribution of the contents of the peers  $P_1, P_2, P_3$  on the cells. Many strategies may be used for mapping the peers to the cells. We may map peer to a cell if it has at least one document in the cell, meaning that the threshold value is set to 0. Alternatively, we map a peer to a cell only if it has a significant number of documents in the cell. The first strategy tends to spread a peer's content over a large number of cells, resulting in the placement of a peer in several clusters. The second strategy on the other hand will place a peer in a reduced number of cells with high density relative to the peer. It also limits the number of clusters a peer belongs to, furthermore reducing time the management overhead of a peer. However, this strategy could fail to find all the documents that match a query due to missing peers in the matched cells.

Cell granularity represents the size of a cell viewed as a sub region of the feature space. Cell granularity and the choice of the threshold  $T$  have a great impact on the efficiency of clustering and on query processing performance. Low cell granularities tend to generate large number of cells. If the actual data distribution of the peer follows a uniform distribution, each peer will equal chance of being mapped to any cell of the feature space. On the other hand, if the actual data distribution follows a spiked distribution (e.g. Zipf or power tail distributions), the low cell granularities create large number of low density cell and the peers are clustered on a few regions of the feature space (regions are discussed further below). A high cell granularity, in the extreme cases, maps all the peers to a single cell. Note that the lower the cell granularity the more accurate the similarity between the documents, and indirectly the peers, which are mapped to a cell. The threshold value will have an impact on query processing, particularly for query that must retrieve all matching documents as discussed above.

**Peer similarity and cluster representation**: We propose a similarity of peer, called *Cell-similarity*. It is defined as follows:

**Definition 1** (*Cell-similarity*)

- Two peers  $P_i$  and  $P_j$  are cell-similar with respect to a cell

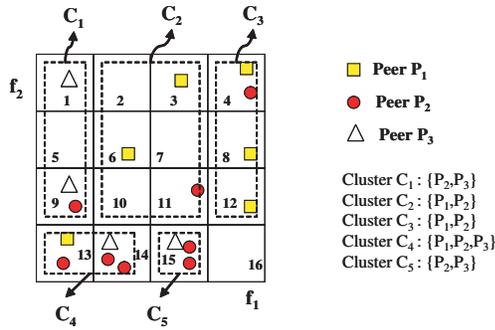


Fig. 3. Partition cells and Clusters

$\varphi \in \phi$  if they are both mapped to  $\varphi$ .

• This definition can easily be extended to define the cell-similarity of two peers  $P_i, P_j$  over a set  $S \subset \phi$ , of cells  $S = \{\varphi_1, \varphi_2, \dots, \varphi_r\}$ . Two peers are cell-similar over a set  $S$  if they are cell-similar with respect to all the cells in  $S$ . □

In figure 3, peers  $P_2, P_3$  exhibit cell-similarity on the cells  $\varphi_{14}, \varphi_{15}$

To define a cluster, we consider a partitioning region of the feature space and group together all peers that are mapped to some cells of the region. A partitioning region, which is a connected region, consists of a set of cells in which each cell is adjacent to other cells of the region. A region-based partition of the feature space is defined as follows:

### Definition 2 (Region-based partition)

Consider  $\phi$  the set of partition cells and  $2^\phi$  the powerset  $\phi$ . Let  $R \subset 2^\phi$  be the set of partitioning regions. A region based partition  $R' \subset R$  is such that  $R' = \{r_i \mid r_i \in R, \bigcup r_i = \phi, \text{ and } \forall (i, j) r_i \cap r_j = \emptyset\}$  □

A cluster  $C_i$  is defined for an element  $r_i$  of a region based partition  $R'$  by including in  $C_i$  peers that are mapped to some connected cells of  $r_i$ . A cluster is defined as:

### Definition 3 (Feature-based cluster)

Given a region based partition  $R' = \{r_1, r_2, \dots, r_k\}$ , a cluster  $C$  is defined as  $C = \{\text{peer } P_i \mid \exists r_j \in R' \text{ and } \forall \varphi_p \in r_j, P_i \text{ is mapped to } \varphi_p\}$  □

Note that different strategies can be used for placing peers in the cluster. One strategy is to include a peer in a cluster if it maps to any cell of the partitioning region on which the cluster is defined. Alternatively, a peer can be placed in a cluster if it belongs to all the cells of partitioning region. Figure 3 depicts 5 clusters on the cells defined above. Peers  $P_1$  and  $P_3$  are in cluster  $C_4$  but are not cell-similar even if each is cell-similar to  $P_2$ . Both  $P_1$  and  $P_3$  can be used to process range queries submitted to cluster  $C_4$ . Cluster  $C_1$  includes the empty cell

$\varphi_5$ . To process queries corresponding to this cell, any query model of the underlying flat unstructure overlay network can be used.

**Signature of a cluster:** A cluster  $C_k$  is represented by a signature vector which summarizes its content. The signature vector  $S(C_k)$  of a cluster  $C_k$  is given by:

$S(C_k) = [\beta_{1k}, \beta_{2k}, \dots, \beta_{jk}, \dots, \beta_{mk}] = \sum_{i=1}^r S_{k_i}$  where  $S_{k_i}$  is the cumulative sum of the peers mapped to the cell  $\varphi_{k_i}$

Given a query  $Q$ , for example a feature based range query, it can be represented by a vector  $Q = [q_1, q_2, \dots, q_j, \dots, q_m]$  Note that if  $C_i \times Q^T = 0$ , the peers of cluster  $C_i$  do not contain multimedia objects corresponding to the query  $Q$ .  $Q^T$  is the transpose of vector  $Q$ , This test can be used to rule the clusters that cannot satisfy a query.

### B. Partitioning features

Another strategy for computing peer similarity is to consider the relative weight or frequency of document occurrences in different range values of a feature. To do so, we divide the range of the feature  $f_i$  into  $m_i$  partitions, where  $m_i = 2^{\delta_i}, \delta_i = 0, 1, 2, \dots$ . Given a peer  $P$ , denote by  $\alpha_k$  the number of documents in  $P$  the  $k^{th}$  partition of the feature  $f_i$ . Some of the values  $\alpha_k$  are zero, corresponding to empty partitions. The content of peer  $P$  over the feature  $f_i$  is given by a vector  $P\vec{f}_i$  defined by:  $P\vec{f}_i = ([\alpha_1, \dots, \alpha_k, \dots, \alpha_{m_i}]^i) / (\sum_{k=1}^{m_i} \alpha_k)$  The signature of  $P$  over all the features is represented by the vector  $P = [P\vec{f}_1, \dots, P\vec{f}_i, \dots, P\vec{f}_n]$ . Similarly, the content of a cluster is represented by a vector  $C = [C\vec{f}_1, \dots, C\vec{f}_i, \dots, C\vec{f}_n]$ , where the  $i^{th}$  component  $C\vec{f}_i$  is defined by:  $C\vec{f}_i = \frac{\sum_{j=1}^{PN} P\vec{f}_{ij}}{PN}$ ,  $PN$  is the number of peers in the cluster and  $P\vec{f}_{ij}$  is the representation vector of the feature  $i$  for peer  $j$ .

Using the above representations, we define a distance measure and use it to group in the same cluster the peers which have similar feature distributions. The distance between a peer  $P$  and a cluster  $C$  is given by  $Distance(P, C) = \sum_{i=0}^n (\sum_{j=0}^{m_i} |P_{ij} - C_{ij}|)$

To illustrate the similarity calculation, assume 3 features are used to describe the content of a peer  $P1$  and two clusters  $C1$  and  $C2$ . Consider the representation  $P = [[0.25, 0, 0, 0.75], [0.875, 0.125], [0, 1, 0, 0]]$  of a peer  $P1$ , and the representations of cluster  $C1$  as  $[[0.529, 0.471, 0, 0], [1, 0], [0.647, 0, 0.353, 0]]$ , and  $C2$  as  $[[1, 0, 0, 0], [0.6, 0.4], [0, 1, 0, 0]]$ . Since  $Distance(P, C1) = 3.75$ ,  $Distance(P, C2) = 2.05$ ,  $P$  is more similar to  $C2$  than  $C1$ .

### C. Clustering in semantic overlay

As discussed in section 2, Crespo et al [4] present a clustering method using classification hierarchy to define semantic overlay networks (SON). Similarly to this approach, we cluster peers according to their semantic features described by a taxonomy.

A taxonomy is a set of concepts  $\Gamma = [C_1, C_2, \dots, C_n]$  related by hierarchical relationships. A document  $D$  is represented by a signature vector  $D[W_1, W_2, \dots, W_n]$ , where  $W_i$  is the

number of occurrences of  $C_i$  in  $D$ . The concept relationships of the taxonomy are taken into account in the computation of the signature vector as follows. Starting from the leaf nodes and moving up to the root of the taxonomy, the weights of the child nodes are added to the weights of their parent. Thus, the weight of a node is updated by the following expression  $w_i^{new} = w_i^{old} + \sum w_j$ . The signature of a document is normalized by  $w_i = \frac{w_i}{\sum_{i=0}^N w_i}$ . The signature of a peer  $P = [Pw_1, Pw_2, \dots, Pw_n]$  is composed of signatures of its documents  $D_i$ , where:  $Pw_i = \frac{1}{DN} \sum_{j=1}^{DN} W_{ij}$  where  $DN$  is the number of documents in the peer. Thus, the signature signature allows the classification of peers into one or more concepts based on the distribution of their contents over the concept hierarchy of the taxonomy. Similar to the feature based clustering method, a peer can be mapped on a concept if the corresponding weight  $W_i$  in the signature vector is greater than a defined threshold  $\rho$ . This classification strategy reduces the number of concepts to which a peer belongs.

Although peers can use several taxonomies to describe their content, in our work in each semantic overlay, there is only one taxonomy. Peers using the same taxonomy will be registered in the same overlay. A cluster is defined as a set of concepts  $\Omega = \{C_1, C_2, \dots, C_m\}$ , where  $\Omega \in \Gamma$ . Each cluster is represented by a mask describing its constituting concepts. For example, if we have a taxonomy of 10 concepts:  $\Gamma = \{C_1, C_2, \dots, C_{10}\}$ . A cluster comprising the concepts  $C_4, C_9$  and  $C_{10}$  has the mask  $[0,0,0,1,0,0,0,1,1]$ . Peers which are classified to  $C_4, C_9$  and  $C_{10}$  will join the cluster. An issue is how to select the concepts used to define the clusters. In our method, the relationships of the concept taxonomy are the basis for defining the clusters. Two concepts  $C_i$  and  $C_j$  are related if  $C_i$  is a sibling, parent or child of  $C_j$ . A cluster is a set of related concepts  $\Omega = \{C_1, C_2, \dots, C_m\}$ , where  $\Omega \in \Gamma$  and  $\forall C_i \in \Omega, \exists C_j \mid C_i$  is related to  $C_j$ .

## V. CACHING IN HON-P2P

Various caching techniques have been proposed in the literature [2], [3], [6], [13], [33]–[35] to improve the performance of multimedia application by reducing query response delays. We propose in this paper a cache method for an efficient information retrieval in HON-P2P. The proposed method generalizes the semantic caching schema presented in [35]. The multimedia cache is based on *Multimedia Segments*, consists of a tuple  $\langle S_Q, S_P \rangle$ .  $S_Q$  is either a semantic or a feature-based query while  $S_P$  represents peers reply to the query  $S_Q$ . Thus, we distinguish two types of multimedia cache: semantic multimedia cache that contains semantic multimedia segments and assigned to a semantic cluster, while feature-based multimedia cache is composed of feature-based multimedia segments and assigned to a feature-based cluster.

A query is represented by a set of multimedia features and can be decomposed into two types of subquery: one is low level feature-based and the other is semantic-based. A query  $Q = \langle Q_L, Q_S \rangle$  is processed by submitting each subquery to the corresponding multimedia cache and combining the partial results to form the final result  $Q$ . When a cache receives a

query  $Q_{i,i=S,L}$ , it searches the stored multimedia segments for similar queries. If there is a cache hit, the relevant peers  $\{P_i\}$  able to answer the query  $Q_i$  are retrieved from the selected cache segments and sent to requesting peer to allow downloading of the requested data. If there is a cache miss, any P2P query processing model is used to locate relevant peers and answers. Once the queried data is found, the requesting peer stores in the cache the multimedia segment containing  $S_{Q_i}$  which represents the query  $Q_i$  and the set of peers  $S_P$  giving answers to the query  $S_{Q_i}$ .

Three strategies can be used for the multimedia cache distribution design. The first is to place the cache in the super peer. It is used to store local queries. External queries are processed and cached by the super peers of other clusters. This type of cache is easy to manage and helps sharing inside clusters the entire query traces between peers which help the information retrieval. However, this is a type of centralized cache which may increase the load of the super peer and require a large storage space.

In the second strategy, each peer maintains a cache in which it keeps the trace of its local queries. The local caches are totally independent, which can generate redundancies. For instance, if a peer P1 processes a query Q1 and stores the results in its own cache and another peer P2 receives a similar query Q2, P2 will not have any access to the related information in the cache of P1 and will duplicate the processing cost of the initial query Q1.

Finally, rather than using the cache in only one level (super peer or peers) a more desirable approach is to distribute the cache over several peers. Whenever a new query is issued, the peers are searched to determine if the query can be answered from the pre-cached answers. This will help reducing the load on the super peer and improve the collaboration between peers. To distribute the cache, we adapt the caching approach proposed in [32]. The cache is distributed over the non empty cells. Each cell is assigned to a peer containing the cache and called an *active peer*. The other peers of the cluster which do not participate in the cache management are called *passive peers* and are registered with the active peers of the cells. When a query  $Q [f_1, f_2, f_j, \dots, f_n]$  is initiated by a peer, it is mapped to a single cell  $\varphi$  called the target cell and served from the appropriate cache. If there is a miss, the active peer of the target cell floods the query to all peers passive registered with him. If there is no result, the query is sent to adjacent clusters. This process is continued until all the appropriate documents are found in the overlay network.

## VI. CONCLUSION

We have presented in this paper a Hybrid Overlay Network HON-P2P consisting of clustering peers taking into account the semantic and feature-based characteristics of their multimedia content. We have defined a weighted concept based method to build semantic clusters, and two methods that use to summarize the low feature representation of peers and cluster peers according to the similarity between representations. Moreover, we have proposed a multimedia cache technique

to improve the research retrieval in HON-P2P network. Two types of multimedia cache are distinguished for each overlay type: semantic multimedia cache and feature-based multimedia cache.

Our ongoing work in the HON-P2P project focuses on (1) addressing the multimedia cache distribution in both semantic and feature-based overlay, (2) defining query processing based on the proposed multimedia cache, and (3) evaluating the performance of our approach on practical cases.

## REFERENCES

- [1] A. Bestavros and S. Mehrotra. Dns-based internet client clustering and characterization. Technical Report BUCS-2001-012 MA 02215, Boston University, Computer Science Department, Boston, June 2001.
- [2] A. Abonamah, A. Al-Rawi, and M. Minhaz. A unifying web caching architecture for the www. *IEEE ISSPIT*, pages 82–94, 2003.
- [3] A. Chankhunthod, P. Danzig, C. Neerdaels, M. Schwartz, and K. Worrell. Hierarchical internet object cache. *Proceedings of the USENIX Technical Conference*, 1996.
- [4] A. Crespo and H. Garcia-Molina. Semantic overlay networks for p2p systems. Technical report, Computer Science Department, Stanford University, October 2002.
- [5] A. Agrawal and H. Casanova. Clustering hosts in p2p and global computing platforms. In *Proceedings of the 3rd International Symposium on Cluster Computing and the Grid (CCGRID)*, pages 367–373, 2003.
- [6] A. Armon and H. Levy. Cache satellite distribution systems: Modeling and analysis. In *IEEE INFOCOM*, 2003.
- [7] Avaki. <http://www.avaki.com/>, 2001.
- [8] S. Bergamaschi and F. Guerra. Peer-to-peer paradigm for a semantic search engine. In *Lecture Notes in Computer Science*, 2530:81–86, 2003.
- [9] A. Crespo and H. Garcia-Molina. Routing indices for peer-to-peer systems. In *Proceedings of the 22nd International Conference on Distributed Computing Systems*, pages 23–32, 2002.
- [10] F.M. Cuenca-Acuna, C. Peery, R.P. Martin, and T. Nguyen. Planetp: Using gossiping to build content addressable peer-to-peer information sharing communities. In *Proceedings of the 12th IEEE International Symposium on High Performance Distributed Computing*, pages 236–246, 2003.
- [11] eDonkey. <http://www.edonkey2000.com>, 2000.
- [12] Entropia. <http://www.entropia.com/>, 2004.
- [13] L. Fan, P. Cao, J. Almeida, and A. Z. Broder. Summary cache: a scalable wide-area web cache sharing protocol. *IEEE/ACM Transactions on Networking*, 8(3):281–293, 2000.
- [14] Gnutella. <http://www.gnutella.com>, 2003.
- [15] GROOVE. <http://www.groove.net>, 2004.
- [16] S. B. Handurukande, A.-M. Kermarrec, F. Le Fessant, and L. Massoulié. Exploiting semantic clustering in the edonkey p2p network. In *Proceedings of the 11th ACM SIGOPS European Workshop*, 2004.
- [17] I. Clarke, O. Sandberg, B. Wiley, and T.W. Hong. Freenet: A distributed anonymous information storage and retrieval system. In *Lecture Notes in Computer Science*, 2009:311–320, 2001.
- [18] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. *ACM SIGCOMM*, pages 149–160, 2001.
- [19] JXTA. <http://www.jxta.org/>, 2004.
- [20] KAZZA. <http://www.kazaa.com/>, 2002.
- [21] B. Krishnamurthy and J. Wang. On network-aware clustering of web clients. *Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 97–110, 2000.
- [22] K. Sripanidkulchai, B. Maggs, and H. Zhang. Efficient content location using interest-based locality in peer-to-peer systems citation. *INFOCOM*, 2003.
- [23] Y. Liu, Z. Zhuang, L. Xiao, and L. M. Ni. Aoto: Adaptive overlay topology optimization in unstructured p2p systems. *Proceedings of IEEE GLOBECOM*, 2003.
- [24] A. Loser, W. Nejdl, M. Wolpers, and W. Siberski. Information integration in schema-based peer-to-peer networks. In *Proceedings of the 15th Conference On Advanced Information Systems Engineering*, 2003.
- [25] S. Marti, P. Ganesan, and H. Garcia-Molina. Sprout: P2p routing with social networks. *International Conference on Extending Database Technology (EDBT)*, pages 425–435, 2004.
- [26] D.S. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja1, J. Pruyne, B. Richard, S. Rollins 2, and Z. Xu. Peer-to-peer computing. Technical Report HPL-2002-57, HP Laboratories Palo Alto, March 2002.
- [27] Napster. <http://www.napster.com/>, 2003.
- [28] W. Nejdl, M. Wolpers, W. Siberski, A. Loser, I. Bruckhorst, M. Schlosser, and C. Schmitz. Semantic overlay clusters within super-peer networks. In *Proceedings of the International World Wide Web Conference*, pages 33–47, 2003.
- [29] W. Nejdl, M. Wolpers, W. Siberski, C. Schmitz, M. Schlosser, I. Bruckhorst, and A. Loser. Super-peer-based routing and clustering strategies for rdf-based peer-to-peer networks. *Proceedings of the twelfth international conference on World Wide Web*, pages 536–543, 2003.
- [30] .NET. <http://www.microsoft.com/net/>, 2004.
- [31] C. Hang Ng and K. Cheung Sia. Peer clustering and firework query model. *Proceedings of 11th World Wide Web Conference*, 2002.
- [32] O. Sahin, A. Gupta, D. Agrawal, and A. El Abbadi. A peer-to-peer framework for caching range queries. *20th International Conference on Data Engineering (ICDE2004)*.
- [33] S. Paul and Z. Fei. Distributed caching with centralized control. *Computer Communications journal*, 24(2):256–268, 2001.
- [34] D. Povey and J. Harrison. A distributed internet cache. *Proceedings of the 20th Australian Computer Science Conference*, pages 175–184, 1997.
- [35] Q. Ren and M. H. Dunham. Semantic caching and query processing. Technical Report 98CSE -04, Southern Methodist University, Dept. of Computer Science and Engineering, May 1998.
- [36] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, pages 329–350, 2002.
- [37] JXTA Search. <http://search.jxta.org/>, 2004.
- [38] SETI@Home. <http://setiathome.ssl.berkeley.edu/>, 2001.
- [39] K. Cheung Sia, C. Hang Ng, C. Hang Chan, S. Kong Chan, and L. Yin Ho. Bridging the p2p and www divide with discover - distributed content-based visual information retrieval. *The Twelfth International World Wide Web Conference (WWW)*, 2003.
- [40] T. Sikora. The mpeg7 visual standard for content. *IEEE Transactions on Circuits and Systems For Video*, 11(6):696–702, 2001.
- [41] I. Tatarinov, Z. Ives, J. Madhavan, A. Halevy, D. Suciu, N. Dalvi, X. Dong, Y. Kadiyska, G. Miklau, and P. Mork. The piazza peer data management project. *SIGMOD Record*, 3:47–52, 2003.
- [42] H. J. Wang, Y. Hu, C. Yuan, Z. Zhang, and Y. Wang. Friends troubleshooting network: Towards privacy-preserving, automatic troubleshooting. *IPTPS04*, 2004.