iCluster: a Self-Organizing Overlay Network for P2P Information Retrieval

Paraskevi Raftopoulou and Euripides G.M. Petrakis

Department of Electronic and Computer Engineering,
Technical University of Crete (TUC),
Chania, Crete, GR-73100, Greece
{paraskevi,petrakis}@intelligence.tuc.gr

Abstract. We present iCluster, a self-organizing peer-to-peer overlay network for supporting full-fledged information retrieval in a dynamic environment. iCluster works by organizing peers sharing common interests into clusters and by exploiting clustering information at query time for achieving low network traffic and high recall. We define the criteria for peer similarity and peer selection, and we present the protocols for organizing the peers into clusters and for searching within the clustered organization of peers. iCluster is evaluated on a realistic peer-to-peer environment using real-world data and queries. The results demonstrate significant performance improvements (in terms of clustering efficiency, communication load and retrieval accuracy) over a state-of-the-art peer-to-peer clustering method. Compared to exhaustive search by flooding, iCluster exchanged a small loss in retrieval accuracy for much less message flow.

1 Introduction

Information sharing in a peer-to-peer (p2p) network requires searching in a distributed collection of peers [1]. Distributed Hash Tables (DHTs) [2, 3] and Semantic Overlay Networks (SONs) [4, 5] are common solutions to the problem of fast information search in p2p networks. DHTs provide fast lookup mechanisms facilitating information search over the network assuming that each peer is connected to other peers and is responsible for a part of the distributed index. SONs provide an alternative solution to the problem of decentralized indexing by relaxing the requirement of strict peer connectivity imposed by DHTs: peers are virtually linked together (forming clusters) based on the likelihood to contain similar content. The problem of finding the most relevant resources is then reduced to the one of locating clusters of peers similar to the query.

We present iCluster, an approach towards efficient organization of p2p networks into SONs that supports Information Retrieval (IR) functionality: iCluster is automatic (requires no intervention by the user), general (requires no previous knowledge of the peers' contents and works for any type of text contents), adaptive (adjusts to dynamic changes of the network contents), efficient

(query processing is faster than existing solutions in the literature) and accurate (achieves high recall outperforming current approaches).

Recent work on SONs by Loser [6] suggests combining information from all layers for scoring the peers. Spripanidkulchai [7] introduced the notion of peer clustering based on similar interests rather than similar documents. Edutella [8] uses metadata to arrange super-peers into the so called *HyperCup* topology [9]. Finally, Lu [10] suggests using content-based information to route query messages to a subset of neighboring peers. However, the work referred above assumes one interest per peer (peer specialization). Klampanos [11] proposed an approach for clustering peers holding information on more that one topics. Parreira [12] introduces the notion of "peer-to-peer dating" for allowing peers to decide which connections to create and which to avoid, based on various usefulness estimators. Additional work on peer organization using SONs is based on the idea of "small world networks" [13, 14]. Schmitz [5] assumes that peers share concepts from a common ontology, and this information is used for organizing peers into communities (small worlds) with similar concepts.

iCluster extends the idea of peer organization in small world networks by Schmitz [5] in the following ways: (a) Peers contribute documents in the network (rather than concepts from an ontology). To that end, peers are represented in the network by their interests (in fact document descriptions derived from their content by automatic text processing). Accordingly, query processing imposes document search operations over the network. (b) This organization allows for peers with multiple and dynamic interests (not known in advance). (c) iCluster proposes new rewiring protocols for achieving dynamic (on the fly) organization of peers in clusters and also, effective information search in the derived clustered organization of peers.

The rest of this paper is organized as follows: iCluster architecture and protocols are discussed in Sect. 2, experimental results are presented and discussed in Sect. 3, followed by conclusions and issues for future research in Sect. 4.

2 iCluster

Each peer is characterized by the content of the documents it contributes to the network. Peers with similar interests are grouped together into clusters. Peers may have more than one interests and belong to more than one clusters. Each peer maintains a *routing index* holding information for short- and long-range links to other peers:

short-range links correspond to *intra-cluster* information (i.e., links to peers with similar interests).

long-range links correspond to *inter-cluster* information (i.e., links to peers belonging to different clusters and thus, having different interests).

Entries in the routing index are of the form $(ip(p_j), c_{jk})$, where $ip(p_j)$ is the IP address of peer p_j the link points to and c_{jk} is the k-th interest of p_j. The number of routing indices maintained by a peer equals the number of peer's

interests. Peers may merge or split their interests by merging or splitting their corresponding routing indices.

2.1 Peer Similarity

Initially, each peer organizes its documents into groups by applying a document clustering algorithm [15]. The documents of a peer may belong to more than one clusters (i.e., the peer may have more than one interests). Documents are represented by term vectors, and each cluster k, $k \in [1, L_i]$, is represented by its centroid c_{ik} (i.e., the mean vector of the vector representations of the documents it contains). Each peer p_i is represented by the list $\{c_{i1}, c_{i2}, \ldots, c_{iL_i}\}$ with the cetroids of its clusters.

A peer p_i can be related to another peer p_j by virtue of more than one interests. The similarity between peers p_i and p_j with respect to interest k of p_i is defined as

$$\mathsf{S}_{ij}^k = S^k(p_i, p_j) = max_{\forall y}\{Sim(c_{ik}, c_{jy})\}, \tag{1}$$

where c_{ik}, c_{jy} are the interests of p_i and p_j, and $Sim(c_{ik}, c_{jy})$ is the similarity between their centroid document vectors. The overall similarity between two peers is defined as the maximum similarity over all pairs of cluster centroids:

$$S(p_i, p_j) = max_{\forall x,y}\{Sim(c_{ix}, c_{jy})\} \tag{2}$$

Finally, the similarity between a document (or query) d and a peer p_i is defined as the maximum similarity between the document (or query) and the peer's interests (centroids):

$$sim(d, p_i) = max_{\forall x}\{Sim(d, c_{ix})\}. \tag{3}$$

2.2 iCluster Protocols

The main idea behind iCluster is to let peers self-organize into SONs, and then, search for similar answers (documents) by addressing the most similar clusters to a given query. The protocols regulating peer join, generation of peer clusters, and query processing in iCluster are discussed next.

Peer Join: When a peer p_i connects to the network, it computes its description $\{c_{i1}, c_{i2}, \ldots, c_{iL_i}\}$. For each interest c_{ik} in its description, p_i maintains a routing index RI_{ik}, which is constructed as follows: p_i issues a request to the network that, through a random walk, collects in RI_{ik} the IP addresses and descriptions from λ (randomly) visited peers, which form the initial neighborhood ν_{ik} of p_i. These (randomly selected) links will be refined according to p_i's k-th interest, using the peer organization protocol below.

Peer Organization: Peer organization proceeds by establishing new connections and by discarding old ones, producing this way groups of peers with similar interests. Each peer p_i periodically initiates a *rewiring procedure*. For each interest k, p_i computes the intra-cluster similarity NS_{ik} (as a measure of cluster cohesion) as

$$NS_{ik} = \frac{1}{|\nu_{ik}|} \cdot \sum_{\forall p_j \in \nu_{ik}} \mathsf{S}_{ij}^{k}, \tag{4}$$

where $|\nu_{ik}|$ is the number of peers in the neighborhood of p_i with respect to interest k. If NS_{ik} is greater than a threshold θ (θ is user defined), then p_i does not need to take any further action, since it is surrounded by peers with similar interests. Otherwise, p_i initiates a cluster refinement process by issuing FINDNODES$= (ip(p_i), c_{ik}, P, t_F)$ message, where $ip(p_i)$ is the IP address of p_i, c_{ik} is the centroid corresponding to k-th interest of p_i and t_F is the time-to-live (TTL) of the message (t_F is user defined). A peer p_j receiving the message computes the similarity between its interest c_{jy} with interest c_{ik} in FINDNODES message, appends to P the interest resulted in the maximum similarity value, reduces t_F by 1 and forwards FINDNODES message to its neighbors. When $t_F = 0$, FINDNODES message is sent back to the initial sender p_i. The message is forwarded with equal probability either to (i) a number m of randomly chosen peers contained in p_j's routing index, or (ii) to the m peers most similar to p_i (the sender of the message). The rationale of applying both forwarding solutions at the same time is not only to connect p_i directly to similar peers, but also indirectly, by enabling propagation of the forwarding message to other similar peers through non-similar peers in the neighborhood of p_i. Figure 1 summarizes the steps of the above rewiring process.

A peer p_j receiving FINDNODES message collects information about new peers with similar interests, and appends it in its routing index $RI_{j\kappa}$ by replacing old short-range links corresponding to less similar peers with new links corresponding to more similar peers. Additionally, p_j collects information about peers with non-similar interests in $RI_{j\kappa}$ updating its long-range links.

Query Processing: Queries are issued by free text or keywords and are formulated as term vectors. The peer issuing the query initiates a QUERY$= (q, t_q)$ message, where q is the query vector and t_q is the TTL of the message (t_q is user defined). The initiator p_i of the message compares q against its interests and decides for the forwarding of the message to some or all of its neighbors according to the *query routing strategy* that follows. Similarly, peers receiving a QUERY message compare q against their interests and forward the message to neighboring peers.

A forwarding peer p_j compares q against its interests and forwards q to its short-range links (i.e., *broadcasts* the message to its neighborhood) if $sim(q, p_j) \geq \theta$. Otherwise, p_j forwards q to m peers, that are the most similar peers to q (*fixed* forwarding). At each step of the forwarding procedure, t_q is reduced by 1.

Procedure Rewiring(p_i, k, t_F, θ, m)
A procedure initiated by a peer p_i whenever its neighborhood similarity NS_{ik} drops below a predefined threshold θ.

input: peer p_i with interest c_{ik} and routing index R_{ik}
output: updated routing index R_{ik}

1: compute $NS_{ik} = \frac{1}{|\nu_{ik}|} \cdot \sum_{\forall p_j \in \nu_{ik}} \mathsf{S}_{ij}^k$
2: **if** $NS_{ik} < \theta$ update routing index R_{ik} as follows
3: $P = \{\ \}$
4: initiate message FindNodes= $(ip(p_i), c_{ik}, P, t_F)$
5: issue FindNodes to neighbors p_j, $j = 1, \ldots, m$, of p_i
the issuing neighbors are with equal probability
m random or the m most similar to p_i
6: let $c_{j\kappa}$ the interest of p_j most similar to c_{ik}
7: $P = P \cup \{(ip(p_j), c_{j\kappa})\}$
8: reduce message TTL t_F by 1
9: **do** the same for the neighbors of p_j
10: **repeat** until message TTL $t_F = 0$
11: return list P to p_i

Fig. 1. Peer organization procedure.

Apart from query forwarding, each peer p_j receiving q applies the following procedure for retrieving documents similar to q. The peer compares q against its interests and if $sim(q, p_j) \geq \theta$ the peer matches q against its locally stored content to retrieve similar documents. Pointers to these documents are sent to the initiator of the query p_i. When this process is completed, p_i produces a list R with results of the form $\langle d, Sim(q, d)\rangle$, where d is a pointer to a document and $Sim(q, d)$ is the similarity between q and d. The candidate answers are ordered by similarity to the query and returned to the user. Figure 2 summarizes the steps of the query processing algorithm.

2.3 Discussion

iCluster is highly dynamic as it allows for random insertions or deletions of new documents in existing peers. Peers recompute their interests when their document collection has fairly changed. iCluster is based solely on local interactions, requiring no previous knowledge of the network structure or of the overall content in the network. Each peer initiates a rewiring procedure every time the overall similarity of the peers in its neighborhood (intra-cluster similarity) drops below a predefined threshold. The cost of this organization results in extra message traffic, which increases with threshold θ. However, this extra message traffic is traded for faster and more efficient search at query time.

Algorithm Query_Routing(QUERY, p_i, t_q, θ, m)
A peer p_i compares the query q towards its interest, finds similar documents and forwards the query to its neighbors.

input: query q issued by peer p_i and threshold θ,
output: document answer set R

1: search within the interests of p_i
2: **if** $sim(q, p_i) > \theta$ **then**
3: $R_i = \{\ \}$
4: search for similar documents within p_i
5: **if** $Sim(q, d) > \theta$ **then**
6: include d in answer set $R_i = R_i \cup (d, Sim(q, d))$
7: **if** $sim(q, p_i) > \theta$ **then**
8: forward q to all short-range links of p_i by issuing Query_Routing
9: **else**
10: forward q to the m neighbors of p_i most similar to q by issuing Query_Routing
11: reduce query TTL t_q by 1
12: search similarly within each visited peer p_j
13: **repeat** until query TTL $t_q = 0$
14: return answer sets R_j to p_i
15: rank results $R = \cup R_j$ on p_j by similarity with q

Fig. 2. Query routing algorithm.

iCLUSTER maintains a fixed number of long-range links (i.e., links to other clusters) in the routing indices of the peers in addition to short-range links. This prevents clusters from becoming isolated and thus inaccessible by other clusters.

Methods such as [8, 5] assuming one interest per peer (specialization assumption) might not perform well under this setting: the description of a peer would either reflect the contents of its strongest interest (e.g., the one with more documents) ignoring all other interests, or result in a single cluster corresponding to the averaging over the entire document collection of the peer. This in turn, would result in poor retrieval performance as queries (even very specific ones) will be addressing highly incoherent clusters of peers. In iCLUSTER, each peer identifies its interests by applying a local clustering process.

3 Evaluation

The experiments are designed to demonstrate the superiority of the proposed iCLUSTER protocols over (a) a state-of-the-art approach for peer organization and retrieval [5] and (b) exhaustive search by flooding [16].

3.1 Experimental Set-Up

iCLUSTER has been tested on a subset of the OHSUMED TREC[1] collection with 30,000 medical articles and on the TREC6[2] data set with 556,078 documents. Each OHSUMED document belongs to one out of 10 classes while, each document in the TREC6 collection belongs to one out of 100 classes.

The network consists of 2,000 peers. Initially, each peer is assigned documents from one class (i.e., initially each peer has one interest). Each peer maintains one routing index with links to other peers (10% are long-range links).

Each peer periodically tries to find better neighbors by initiating the rewiring procedure. The base unit for time used is the period t. The start of the rewiring procedure for each peer is randomly chosen from the time interval $[0, 4K \cdot t]$ and its periodicity is randomly selected from a normal distribution of $2K \cdot t$ for each peer separately. We start checking the network at time $4K \cdot t$, when all peers have initiated at least once the rewiring procedure.

We experimented with different values of similarity threshold θ, message forwarding TTL t_F and query forwarding TTL t_q. In the following, we show how the critical values characterizing the network vary over time. We considered 5 different initial network topologies and for each topology the results were averaged over 5 runs.

3.2 Performance Measures

The performance of iCLUSTER is mainly evaluated in terms of peer organization, communication load and accuracy of retrieval (recall). The (*weighted*) *clustering coefficient* $\overline{w\gamma}$ is one of the common metrics [17, 18] used to describe how well the peers are organized into groups with similar interests and is defined as:

$$\overline{w\gamma} = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{1}{\lambda(\lambda - 1)} \sum_{\forall p_j, p_k \in \nu_i, p_k \in \nu_j} S(p_j, p_k) \right) \tag{5}$$

The *network load* of a method is measured by the number of messages exchanged by the peers during rewiring or querying. In turn, the accuracy of retrieval is evaluated using *recall* (i.e., percentage of qualifying answers retrieved with respect to the total number of qualifying answers in the network). An organization (or search) strategy is better than another if it achieves better clustering coefficient (or retrieval accuracy) for less communication load.

3.3 Peer Organization

To evaluate the clustering effectiveness of iCLUSTER, we monitored how $\overline{w\gamma}$ varies over time for different values of θ and t_F. After a few iterations (after $9K \cdot t$), $\overline{w\gamma}$ stabilizes to 0.21 for the OHSUMED and 0.55 for the TREC6 data set. The

[1] http://trec.nist.gov/data/t9_filtering.html
[2] http://boston.lti.cs.cmu.edu/callan/Data/

variation in $\overline{w\gamma}$ is due to the variation in the number of document classes in the two data sets. Stability is achieved as peers are surrounded eventually by other peers with similar interests (i.e., $NS > \theta$). The experiments demonstrate that the values of $\overline{w\gamma}$ are slightly influenced by θ (less than 3%). Additionally, only a small number of organization messages are initially needed and are reduced to 0 after time $6K \cdot t$, when the network becomes coherent.

Additional experiments indicate that the network converges faster for higher values of t_F (i.e., high values of t_F address peers far apart from the peer originating the process). Although $\overline{w\gamma}$ do not vary significantly with t_F, the communication overhead increases by 86% (i.e., when t_F increases from 4 to 7), leading us to choose $t_F = 4$ for our setting.

The experiments above demonstrate that the rewiring protocol of iCluster results in a effective peer organization at the expense of small communication load. The rewiring similarity threshold θ affects clustering cohesion, while the rewiring TTL parameter t_F has minimal effects on the convergence time of the network.

3.4 Performance of Retrieval

The purpose of this set of experiments is to evaluate the performance of the proposed query routing protocol as a function of (i) recall, (ii) communication overhead incurred by a query and (iii) recall per search message. We also examine the dependance of recall and communication overhead on t_q, θ and t_F. The plots below correspond to measurements on the OHSUMED data set (TREC6 produced similar results).

Figure 3 illustrates how recall varies with time for various values of θ. When no similarity structure is imposed in the network ($4K \cdot t$), the queries are flooded over the network reaching recall as high as 0.8. However, this value of recall is achieved for large communication overhead (1,200 messages per query). When the network becomes organized into cohesive clusters (after time $9K \cdot t$), iCluster achieves the same high values of recall for much less communication overhead (500 messages per query).

As shown in Fig. 3, $\theta = 0.6$ achieves the highest recall on an organized network (after time $9K \cdot t$). For lower values of θ, there are many links from each peer towards non-similar peers, since the clusters are not coherent enough. For higher values of θ, the clusters are coherent but it becomes difficult for a query to be forwarded to similar clusters of peers through other non-similar peers. The optimal value of t_F achieving the best recall is 6.

Figure 4 shows the dependence of recall and communication load incurred by the query (in number of messages sent) on t_q. Obviously, recall increases with t_q as more peers are receiving q, but communication load increases as well. For $t_q = 4$ or 6 the recall achieved is very low. Notice that $t_q = 10$ achieved almost 19% better recall (approaching recall 1) than $t_q = 8$ at the expense of 53% more communication load. Based on these observations, the suggested value of t_q is 8. Although $t_q = 8$ is relatively high, the communication overhead is low as iCluster applies selective propagation of query messages to qualifying peers.

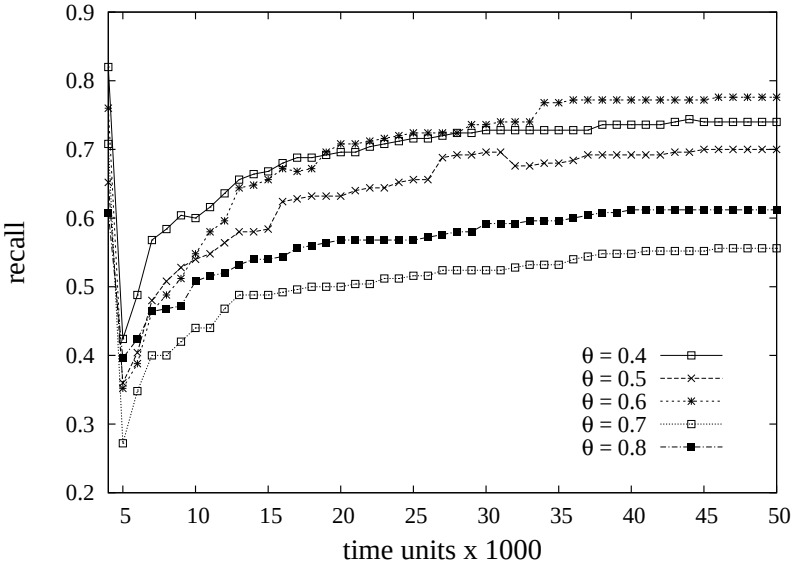

Fig. 3. Recall as a function of time for various values of θ.

The experiments above showed that iCLUSTER achieves high values of recall for less communication load when the network becomes organized into cohesive clusters. We examined the dependance of the retrieval performance on the rewiring process and on t_q, and we suggested optimal values for the parameters (i.e., achieving high recall with small communication overhead).

3.5 Comparison to Other Methods

The following methods are implemented and compared:

- iCLUSTER, the method proposed in this work for $\theta = 0.6$, $t_F = 6$ and $t_q = 8$.
- Query flooding, the method implemented by many p2p systems (e.g., Gnutella). It assumes no special network structure and the query is flooded over the network. For comparison with iCLUSTER we set $t_q = 8$.
- The peer organization approach proposed by Schmitz [5], using $\theta = 0.6$, $t_F = 6$ and $t_q = 8$ to have results that are comparable with iCLUSTER.

Notice that $\overline{w\gamma}$ is close to 0 for flooding, indicating no organization of peers into clusters. Compared to [5], iCLUSTER results in higher value of $\overline{w\gamma}$ (0.21 as opposed to 0.08) and therefore, better clustering quality.

Figure 5 shows how recall varies over time for all three approaches. The flooding approach achieves recall 0.85 as it searches the network almost exhaustively imposing high communication overhead. As Fig. 5 indicates, prior to imposing any similarity structure in the network (before time $6K \cdot t$), iCLUSTER and [5] achieves recall as high as 0.8. However, notice the high communication overhead

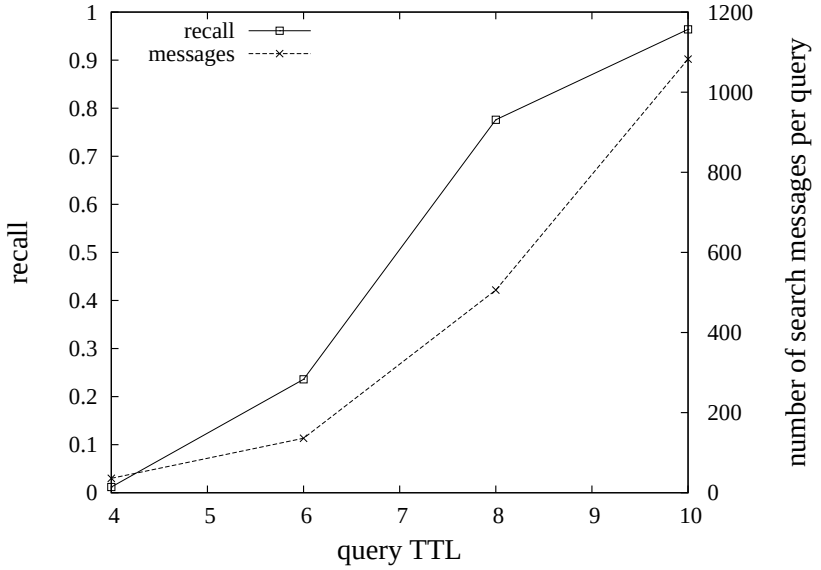

Fig. 4. Recall and query messages per query as a function of t_q.

incurred by both methods (i.e., almost 1,000 messages per query). When the network becomes organized, iCluster (unlike [5]) achieves recall resembling that achieved by the flooding approach but for much less (up to 60%) communication overhead. Finally, Fig. 6 indicates that, in terms of message traffic per query, flooding is the worst method.

In this set of experiments, iCluster is compared with the peer clustering approach by Schmitz [5] and the standard exhaustive search approach by flooding, in terms of both communication load and retrieval accuracy. The experiments showed that iCluster benefited the most from creating coherent clusters of peers with similar interests, as it resulted in high recall for much less network load than all its competitor methods. In particular, iCluster can be almost as effective as flooding for much less message flow (i.e., the communication load reduced by approximately 70%).

4 Conclusion

We present iCluster, an approach for organizing peers into clusters and for supporting information retrieval functionality. iCluster ensures clustering coherence while achieving high accuracy of retrievals by issuing a periodic rewiring procedure. The experimental results demonstrated that iCluster outperforms other approaches for peer organization and retrieval, achieving higher clustering quality and higher recall for less communication overhead. Future work, includes experimentation with different query distributions, the study of the effect

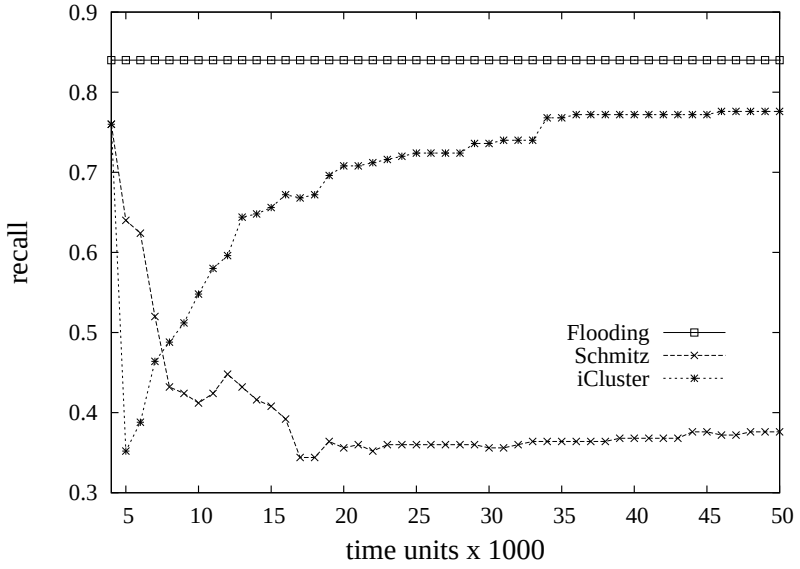

Fig. 5. Recall as a function of time for (a) iCluster (b) Flooding and (c) Schmitz [5].

of churn (dynamic peer insertions/deletions) to network organization and data retrieval, and extension of the proposed protocols to support both information retrieval and filtering functionality (publish/subscribe).

Acknowledgements

This work was funded by project "Herakleitos" of the Greek Secretariat for Research and Technology. We are grateful to Dr. Gerhard Weikum and the members of the Databases and Information Systems group of Max-Planck-Institute for their support.

References

1. Milojicic, D.S., Kalogeraki, V., Lukose, R., Nagaraja, K., Pruyne, J., Richard, B., Rollins, S., Xu, Z.: Peer-to-Peer Computing. Technical report, HP Labs (2002)
2. Stoica, I., Morris, R., Liben-Nowell, D., Karger, D.R., Kaashoek, M.F., Dabek, F., Balakrishnan, H.: Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications. IEEE/ACM Trans. on Networking **11**(1) (2003)
3. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S.: A Scalable Content-Addressable Network. SIGCOMM '01
4. Crespo, A., Garcia-Molina, H.: Semantic Overlay Networks for P2P Systems. Technical report, Stanford Univ. (2003)
5. Schmitz, C.: Self-Organization of a Small World by Topic. P2PKM '04

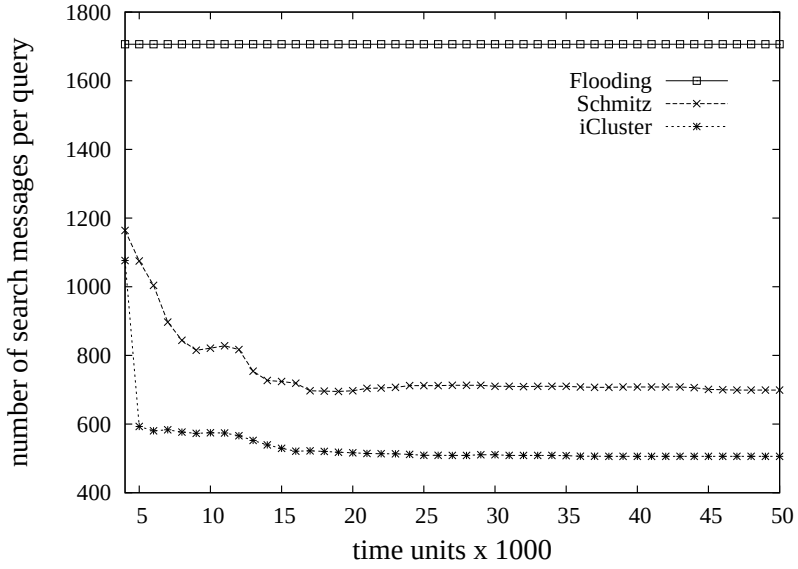

Fig. 6. Number of search messages as a function of time for (a) iCluster (b) Flooding and (c) Schmitz [5].

6. Loser, A., Tempich, C.: On Ranking Peers in Semantic Overlay Networks. WM '05
7. Spripanidkulchai, K., Maggs, B., Zhang, H.: Efficient Content Location using Interest-Based Locality in Peer-to-Peer Systems. INFOCOM '03
8. Nejdl, W., Wolf, B., Qu, C., Decker, S., Sintek, M., Naeve, A., Nilsson, M., Palmer, M., Risch, T.: EDUTELLA: a P2P Networking Infrastructure based on RDF. WWW '02
9. Decker, S., Schlosser, M., Sintek, M., Nejdl, W.: HyperCuP - Hypercubes, Ontologies and Efficient Search on P2P Networks. AP2PC '02
10. Lu, J., Callan, J.: Content-Based Retrieval in Hybrid P2P Networks. CIKM '03
11. Klampanos, I., Jose, J.: An Architecture for Information Retrieval over Semi-Collaborating Peer-to-Peer Networks. ACM SAC '04
12. Parreira, J.X., Michel, S., Weikum, G.: p2pDating: Real Life Inspired Semantic Overlay Networks for Web Search. Inf. Proc. and Manag. **43**(1) (2007)
13. Crespo, A., Garcia-Molina, H.: Routing Indices for P2P Systems. ICDCS '02
14. Li, M., Lee, W.C., Sivasubramaniam, A.: Semantic Small World: An Overlay Network for Peer-to-Peer Search. ICNP '04
15. Steinbach, M., Karypis, G., Kumar, V.: A Comparison of Document Clustering Techniques. TextDM '00
16. Hughes, D., Coulson, G., Walkerdine, J.: Free Riding on Gnutella Revisited: The Bell Tolls? IEEE DS Online **6**(6) (2005)
17. Schmitz, C., Staab, S., Tempich, C.: Socialisation in Peer-to-Peer Knowledge Management. I-KNOW '04
18. Hui, K., Lui, J., Yau, D.: Small-world Overlay P2P Networks: Construction, Mmanagement and Handling of Dynamic Flash Crowds. Computer Networks **50**(15) (2006)