Clustering using Hypergraph for P2P Query Routing
Simulation and Evaluation

Anis Ismail[1], Mohammad Hajjar[1], Mohamed Quafafou[2],
Nicolas Durand[2] and Mazen El Sayed[1]
[1]*Lebanese University, Beirut, Lebanon*
[2]*LSIS, Domaine Universitaire de Saint-Jérôme, Marseille, France*

Keywords: Ecclat, Hypergraphs, Mtminer, P2P Network, Query Routing.

Abstract: Peer-to-peer overlay networks offer a flexible architecture for decentralized data sharing. In P2P schema-based systems, each peer is a database management system in itself, ex-posing its own schema. In such a case, the main objective is the efficient search across peer databases by processing each incoming query without overly consuming bandwidth. The usability of these systems depends on efficient and effective routing of content-based queries is an emerging problem in P2P networks. This work was attended to motivate the use of mining algorithms in the P2P context to improve the efficiency of such methods. Our proposed method combines clustering and hypergraphs. We use ECCLAT to build approximate clustering and discovering meaningful clusters with slight overlapping. We use the algorithm MTMINER to extract all minimal transversals of a hypergraph (clusters) for query routing. The set of clusters improves the robustness in queries routing mechanism and scalability in P2P Network. Our experimental results prove that our method generates impressive levels of performance and scalability with respect to important criteria such as response time, precision and recall.

1 INTRODUCTION

Peer-to-Peer (P2P) has emerged as an efficient way to share huge volumes of data (Akbarinia and Martins, 2006). The most important problem in such networks is query routing, i.e. deciding to which other peers, the query has to be sent for high efficiency and effectiveness. However, systems that broadcast all queries to all peers suffer from limited effectiveness and scalability.

Nodes, like super-peers, process queries and produce as results, a group of peers; the result of a query is the union of results from every super-peer (SP) and their peers that process the query. When a peer submits a query, this peer becomes the source of this query that is transmitted to its SP. The routing policy in use determines relevant neighbours SP quickly, based on semantic mappings between schemas of (super-)peers, and then send the query to them. When a SP receives a query, it will process it over its local collection of data sources taking into account its different peers. If at least one of its peers answers the query then results are found and the SP will send a single response message back to the peer source. The most important challenge for the information retrieval in P2P networks is also to be able to direct the query to the other peers that contain the most relevant answers in a fast and competent way.

Our main goal is the efficient search across the P2P network while routing the queries directly to relevant peers. To accomplish this goal, it is crucial that each query is not broadcast into the whole network, but is routed to a relevant set of peers. Furthermore, the efficiency and good performance of the whole P2P network does not only depend on how the query is routed to relevant peers, but also on how it is routed to these relevant peers with minimum query processing and bandwidth consumption.

The following section presents some related works. Section 3, presents the baseline algorithm of queries routing in hybrid P2P systems and the concept of hyper-graph. Section 4 presents our work “Minimal covering shortcut” approach. Section 5 presents Experiments and Evaluations. In Section 6, we present the conclusion.

Ismail A., Hajjar M., Quafafou M., Durand N. and El Sayed M..
Clustering using Hypergraph for P2P Query Routing - Simulation and Evaluation.
DOI: 10.5220/0004452302470254
In *Proceedings of the 15th International Conference on Enterprise Information Systems* (ICEIS-2013), pages 247-254
ISBN: 978-989-8565-59-4

2 RELATED WORKS

Knowledge discovery and data mining (KDD) in Peer-to-peer network is a relatively new field with little related literature. P2P data mining has recently emerged as an area of KDD research, specifically focusing on algorithms which are efficient in query routing and scalability. For instance, Bhaduri (Bhaduri et al., 2008) propose an alternate solution that works in a completely asynchronous manner in distributed environments and offers low communication overhead, a necessity for scalability.

Content location is a challenging problem in decentralized peer-to-peer systems. And query-flooding algorithm in Gnutella system suffers from poor scalability and considerable network overhead. Currently, based on the Small-world pattern in the P2P system, a piggyback algorithm called interest based shortcuts gets a relatively better performance.

In P2P systems, research, such as P-Grid (Aberer, 2001) Chord, CAN (Ratnasamy et al., 2001), or is based on various forms of distributed hash tables (DHTs) and supports mappings from keys, e.g., titles or authors, to locations in a decentralized manner such that routing scales well with the number of peers in the system. PlanetP (Cuenca-Acuna et al., 2003) is a publish-subscribe service for P2P communities and the first system supporting content ranking search. PlanetP distinguishes local indexes and a global index to describe all peers and their shared information. The global index is replicated using a gossiping algorithm. The system, however, is limited to a few thousand peers.

Strategies for P2P request routing beyond simple key lookups but without considerations on ranked retrieval have been discussed in (Cohen et al., 2003), (Crespo and Garcia-Molina, 2002), but are not directly applicable to our setting. The construction of semantic overlay networks is addressed in, (Crespo and Garcia-Molina, 2002) using clustering and classification techniques; these techniques would be orthogonal to our approach. Tong et al. (Tong and Yang, 2005) distribute a global index onto peers using LSI dimensions and the CAN distributed hash table. In this approach peers give up their autonomy and must collaborate for queries whose dimensions are spread across different peers. (Aberer et al., 2004) addresses the problem of building scalable semantic overlay networks and identifies strategies for their traversal. Castano and Montanelli addressed the problem of formation of semantic Peer-to-Peer communities (Castano and Montanelli, 2005). Each peer is associated with an ontology which gives a semantically rich representation of the interests that the peer exposes to the network, in terms of concepts, properties and semantic relations. Each peer interacts with others by submitting discovery queries in order to identify the potential members of an interest-based community, and by replying to incoming queries whether it can join a community. A semantic matchmaker is employed to check whether two peers share the same interests. Datta et al. proposed an algorithm for K-Means clustering over large, dynamic networks (Datta et al., 2006).

3 SEMANTIC MAPPINGS AND HYPER-GRAPHS

This section is devoted to the study of two methods developed and used for queries routing in P2P communities. The baseline method developed in (Faye et al., 2007), uses semantic similarity functions to establish semantic mapping between peers and peers/super-peers. Unfortunately, this approach is not being scale due to the mappings it uses and this problem arise considering only thousands of Peers in the network. This limit motivates our investigation and the development of our new method based respectively on clustring/hypergraphs.

3.1 Baseline Approach

A new Peer Pj advertises its expertise by sending, to its Super-Peer, a domain advertisement DA_j = (PID; E^{j}_{XP} , T_j ; $Ɛ_{acc}$; TTL) containing the Peer ID denoted PID, the suggested expertise E^{j}_{XP} , the topic area of interest T_j, the minimum semantic similarity value (Ɛacc) required to establish semantic mapping between the suggested expertise E^{j}_{XP} and the theme of its SP. When receiving an expertise E^{j}_{XP} , a Super-Peer SP_A invokes the semantic matching process to find mappings between its suggested schema and the received expertise.

The semantic routing algorithm (Algorithm 1) of baseline approach exploits the expertise of (super-)Peers and the two levels of mappings in order to forward a query Q to only relevant Super-Peers. A Peer P_2 submits its query Q_2 on its local data schema. This query is sent to his Super-Peer SP_A responsible for the community (See Figure 1). The Super-Peer SP_A in turn suggests, based on the index obtained by the process of mediation (first level), the

Peers P_1 of his community or the other Super-Peers SP_P that are able to treat this query. Each submitted query received by a Super-Peer, is processed by searching connections (second level of mappings) between the subject of this query and expertise of Peers (of the same community) or the description of themes of other Super-Peers.

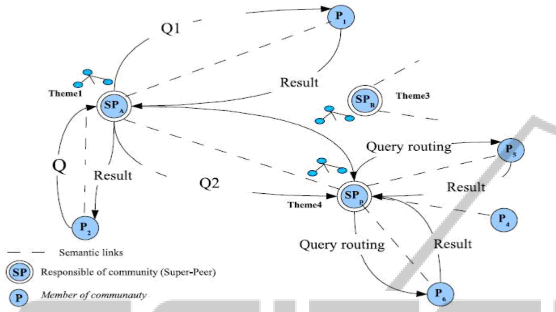

Figure 1: Network configuration and query routing (Baseline approach).

Algorithm 1: Baseline algorithm

```
Input: Q : Query
       SP : Super-peer of P
Output: SRQ : Set of answers of Q

Variables : PSet : Set of peers
 NP : Neighbours of SP (set of super-peers)
SRQ = Φ
PSet = Capacity CMSP/P (Q) >$\varepsilon_{acc}$
repeat
       SPQ = get(s ϵ PSet);
       Remove SPQ from PSet;
       SRQ = SRQ  ∪  Query(SPQ);
 until (PSet = Φ)
 repeat
     SPQ = Capacity CMSP/SP (Q) > $\varepsilon_{acc}$
      Remove SPQ from NP;
      SRQ = SRQ ∪  BL(Q, SPQ);
 until (PSet = Φ)
 Return(SRQ);
```

In turn, a SP from the nearby community, having received this request, researches among Peers (in his community) who are able to answer this query. The major problem of this approach is the mediation at the two levels cited above: if we take thousands of Peers or Super-Peers this approach can not be scale due to the mappings at both levels.

The routing of Query in these networks is therefore very problematic. Semantic Routing is a method of routing which focuses on the nature of the query to be routed than the network topology. Essentially semantic routing improves on traditional routing by prioritizing nodes which have been previously good at providing information about the types of content referred to by the query. Semantic Routing is obviously not the most optimal solution for routing, and it wasn't long before other P2P routing algorithms emerged which were more efficient.

Assuming that peer P_2 issues a query Q_2, the query routing algorithm proceeds as follows:

- We first find the responsible super-peer for P_2 which in this example is SP_A.
- The responsible Super-Peer (SP_A) process the query to find the relevant peers of his community (ex.: P_1) if there are, and also find the others Super-Peers (ex.: SP_P) that might contain relevant peers to answer the query.
- Each relevant Super-peer(s) (SP_A, SP_P) treat(s) query to find relevant peers using the function CAP that measures the capacity of a peer of expertise EXP(P_1) on answering a given query of subject of Sub(Q).

$$Cap(P,Q)=\frac{1}{\mathrm{Sub(Q)}}(\sum_{s\,\in\mathrm{Sub(Q)}} \underset{e\,\in\,\mathrm{Exp(P)}}{Max}\; Ss(s,e))$$

- Then the final set of relevant peers ((P_1:SP_A)...(P_5:SP_P)) and their corresponding super-peers are returned. Semantic routing is not a reasonably idea when the network growth. This motivates us to develop a new approach based on clustering super-peer.

3.2 Hypergraph Transversals based Approach

This section introduces a new efficient method for queries routing in the P2P context that is based on both the super-peer clustering algorithm called Ecclat (Durand and Cremilleux, 2002), (Durand et al., 2006) and the computation of a minimal query routing strategy. The clustering of super-peers using their expertise leads to the construction of communities where each one is represented by a set of super-peers (cluster of super-peers) with the constraint that a super-peer may belong to more than one cluster. In this situation the set of clusters constitutes a set of hypergraph and where each node constitutes a community. The question is than how to find the minimal querying strategies where each one is a set of super-peers that covers all communities. The function cover means that the minimal set contains at least on super-peer of each community. Consequently, this strategy guaranties that one represents all expertise of the network. Thus, we consider that a strategy is a semantic context that can be useful for queries routing. In fact, when a super-peer SP receives a query Q and can not answer it using only its peers then it selects the possible minimal strategy minS where SP ∈ minS.

A transversal is minimal in the sense that guaranties that all communities (clusters of super-peers) are represented:

$$\forall Tc \in T ; \forall c \in C : Tc \cap c \neq \phi ;$$

Where C is the set of communities (super-peers clusters), T is the set of transversals.

In our context, we cluster super-peers according to their expertise. Table1 presents an example of transactional dataset. There are 8 transactions (denoted SP_1… SP_8) and 9 items (denoted W_1… W_9). Transactions correspond to super-peers. Items correspond to components of a query successfully processed by the super-peers. For example, W_1 is present in the transaction SP_1 because W_1 is a component of a query successfully processed by the super-peer SP_1. The obtained clusters with minfr=20% and M=1 (M is an integer corresponding to a number of transactions not yet classified that a new selected cluster must classify) (minfr: minimum number of transactions in a cluster) (Durand, 2002) are: (W_1, W_2, W_3; SP_1, SP_2, SP_3), (W_4, W_5; SP_4, SP_5, SP_6), (W_1, W_6, W_7; SP_6, SP_7) et (W_9; SP_7, SP_8).

Table 1: Example of a dataset D1.

Id.	Items								
SP_1	W_1	W_2	W_3						
SP_2	W_1	W_2	W_3						
SP_3	W_1	W_2	W_3						
SP_4				W_4	W_5				
SP_5				W_4	W_5			W_8	
SP_6	W_1			W_4	W_5	W_6	W_7	W_8	
SP_7	W_1					W_6	W_7		W_9
SP_8								W_8	W_9

The cluster (W_1, W_2, W_3; SP_1, SP_2, SP_3) shows that SP_1, SP_2 and SP_3 share an expertise characterized by the association of the components W_1, W_2 and W_3.

Table 2 presents another example with 300 Peers and 10 Super-Peers. The resulting clusters minfr = 20% and M=1 are:

(W_{19}, W_{37}, W_{40}, W_{41}, W_{45}, W_{46}; SP_5, SP_6, SP_{10}), (W_{17}, W_{36}, W_{37}, W_{38}, W_{39}, W_{41}, W_{42}; SP_4, SP_6, SP_7), (W_6, W_{21}; SP_2, SP_8, SP_9), (W_5, W_6, W_8; SP_1, SP_2, SP_8), (W_2, W_4; SP_1, SP_3, SP_5)

Figure 2 focuses only on the resulted five clusters. An interesting feature of the clustering algorithm is its ability to produce a clustering with a minimum overlapping between clusters (approximate clustering) or a set of clusters with a slight overlapping.

Table 2: A dataset D2.

Id.	Items
SP_1	W_1 W_2 W_3 W_4 W_5 W_6 W_7 W_8 W_9 W_{10} W_{11} W_{12} W_{13} W_{14} W_{15} W_{16} W_{17} W_{18}
SP_2	W_1 W_3 W_5 W_6 W_7 W_8 W_9 W_{10} W_{11} W_{12} W_{14} W_{19} W_{20} W_{21} W_{22} W_{23} W_{24}
SP_3	W_2 W_4 W_9 W_{18} W_{25} W_{26} W_{27} W_{28} W_{29} W_{30} W_{31} W_{32} W_{33} W_{34}
SP_4	W_3 W_{17} W_{24} W_{35} W_{36} W_{37} W_{38} W_{39} W_{40} W_{41} W_{42}
SP_5	W_2 W_4 W_{11} W_{12} W_{19} W_{37} W_{40} W_{41} W_{43} W_{44} W_{45} W_{46}
SP_6	W_1 W_{11} W_{13} W_{14} W_{17} W_{19} W_{24} W_{35} W_{36} W_{37} W_{38} W_{39} W_{40} W_{41} W_{42} W_{45} W_{46} W_{47}
SP_7	W_2 W_6 W_{11} W_{17} W_{20} W_{36} W_{37} W_{38} W_{39} W_{41} W_{42} W_{43} W_{44} W_{48} W_{49} W_{50} W_{51} W_{52} W_{53}
SP_8	W_5 W_6 W_8 W_{21} W_{23} W_{24} W_{25} W_{28} W_{30} W_{33} W_{44} W_{54}
SP_9	W_6 W_{21} W_{36} W_{42} W_{49} W_{50} W_{52} W_{53} W_{55} W_{56} W_{57} W_{58} W_{59} W_{60} W_{61} W_{62} W_{63}
SP_{10}	W_{19} W_{37} W_{40} W_{41} W_{45} W_{46} W_{47} W_{64} W_{65}

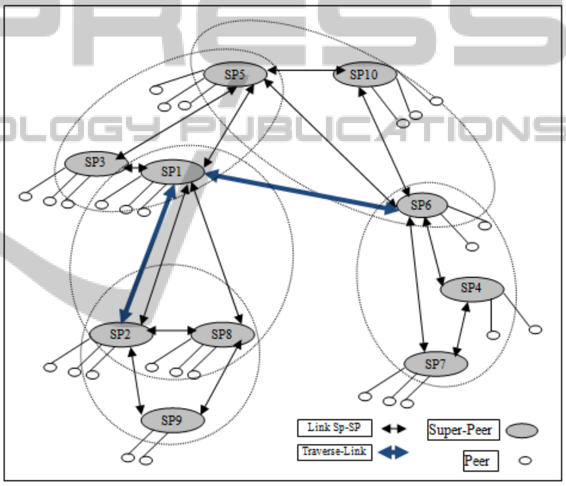

Figure 2: Obtained Clusters with traversals.

4 MINIMAL COVERING SHORTCUT APROACH

The goal of this method is to characterize collectively the communities. Therefore we characterize not a particular community, but all communities in the network. First, we explicit communities with a clustering algorithm. Then, we formalize the problem of collective characterizing of communities as research of MCS (Minimal Covering Shortcuts) are shortcuts between Super-Peers, minimum covers all communities.

Indeed, communities are built automatically using the clustering algorithm ECCLAT (Durand and Cremilleux, 2002). This is done by analyzing the queries answered by each super-peer. Each community is represented by a set of super-peers (cluster of super-peers), with the constraint that a super-peer can belong to more than one cluster. In this situation, the entire cluster is a hypergraph, where each node is a super-peer. The question then is how to exploit such a hypergraph to define strategies for minimal query routing.

We define an MCS as a set of super-peers covering all communities (a collection of super-peers containing a super-peer for each community). The naive method is to take a super-peer of each community and define MCS as the union of all its super-peers. Therefore, each MCS will have as cardinality the number of communities in the network. As communities overlap, we will search MCS sets with minimum size respecting the coverage constraint. Thus, we consider that MCS is a semantic context, which can be useful for routing queries.

Definition (Minimal Covering Shortcut - MCS): Given Ω set of domains, Φ set of communities in the network ($\Phi \subset 2\Omega$), $X \in 2\Omega$, X says MCS if it satisfies the constraints of coverage and the following minimalist: (1) coverage : $\forall\ Y \in \Phi$, $Y \cap X \neq \emptyset$ et (2) minimal : $\forall\ Y \subset X$, $\exists\ Z \in \Phi$,: $Z \cap Y = \emptyset$.

Note that $T\Phi \in 2\Omega$, set of calculated MCS from all communities Φ of network P2P and we say that MCS is associated with a community if it contains at least one member of this community.

For any given community, there is at least one MCS associated with it:

$$\forall_{c} \in \Phi, \exists\, T \in T\Phi: T \bigcap c \neq \phi\ ;$$

Indeed, suppose the contrary: there is a community c such that for all MCS T we have $T \bigcap c = \phi$. This means that there is community c which has no representative in T. This is a contradiction with the property T that coverage must cover all communities (it does not cover c).

However, it is possible that a SP (field) is not in any MCS.

4.1 Architecture of MCS-Super-Peer (MCS-SP)

In this section, we present the architecture of a MCS-SP (Ismail et al., 2010). The general architecture of a MCS-SP is described in Figure 3 each MCS-SP contains the following components:

- Query Manager: The role of this component is to rewrite and route queries to the (Super-)Peers. It also defines the implementation plans and optimizes front to supervise their implementation across the network.
- Communication module: As for the Peer communication, it is provided by Sun's JXTA [JXTA. www.jxta.org]
- Table of MCS: Contains a list of associated community ties.
- k-Filter: Allows you to find k cross ties among community to treat a given query.

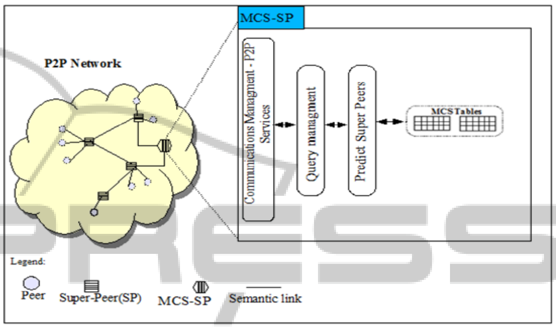

Figure 3: Architecture d'un MCS-SP.

4.2 Calculate of Communities MCS

Clusters (Figure 4) are used to model the P2P network as hypergraph. Each vertex is a SP. A hyperedge corresponds to a cluster. Figure 4 shows an example of the hypergraph. Then we use the minimum traverse of the hypergraph to link all vertices (SP) and thus forming routes for routing queries. The calculation of MCS belongs to the calculation of minimum traverses.

The results of all minimum traversals calculated with MTMINER (Hebert, 2007) are:

Minimum traversals = {{SP_1, SP_2, SP_6}, {SP_1, SP_6, SP_8}, {SP_1, SP_6, SP_9}, {SP_2, SP_3, SP_6}, {SP_2, SP_4, SP_5}, {SP_2, SP_5, SP_6}, {SP_2, SP_5, SP_7}, {SP_3, SP_6, SP_8}, {SP_4, SP_5, SP_8}, {SP_5, SP_6, SP_8}, {SP_5, SP_7, SP_8}, {SP_1, SP_2, SP_4, SP_{10}},{SP_3, SP_7, SP_8, SP_{10}}, {SP_1, SP_2, SP_7, SP_{10}}, {SP_1, SP_4, SP_5, SP_9}, {SP_1, SP_4, SP_8, SP_{10}}, {SP_1, SP_4, SP_9, SP_{10}}, {SP_1, SP_5, SP_7, SP_9}, {SP_1, SP_7, SP_8, SP_{10}}, {SP_1, SP_7, SP_9, SP_{10}}, {SP_2, SP_3, SP_4, SP_{10}}, {SP_2, SP_3, SP_7, SP_{10}}, {SP_3, SP_4, SP_8, SP_{10}}}

This method, searching for communities and calculation of MCS, remains centralized, performed by a central server. Once the calculation of the MCS is made, the central server is responsible for sending for each community the traverses associated with it. These are stored in the MCS-SP.

4.3 Query Routing in the MCS Architecture

Once the communities (clusters) are determined and calculated MCS, we will use them to route queries to a set of super-peers and avoid the spreading to the whole network. We develop what we called routing strategies based on MCS. Thus, when a query Q is sent by the peer P to the super-peer SP, this SP will use the MCS associated with the community to identify super-peers to which the query Q will be sent. Several strategies are possible to achieve this goal:

- **0-Filter:** there is no MCS passed by a SP.
- **1-Filter:** This strategy consist of selecting one of MCS, noted T, which is associated with the community c and containing SP, then send the query Q to all super-peer of T except SP.
- **k-Filter:** This is the generalization of strategy 1-Filter, since we consider k MCS associated with the community c instead of one. Then, the query Q is sent to each k MCS the strategy according to 1-Filter. The union of all results returned by each MCS is the final result of k-Filter.
- ***-Filter:** we use here all MCS associated with the community c.

The following algorithm uses only k MCS (strategy) associated with community c to answer the query Q sent by peer P which the super-peer belongs c (algorithm 2):

Algorithm 2: k-Filter Calculate the Responses of Q using the MCS

```
Input: query Q
Output : RQ, AQ
SP_Q : super-peer that receive the query Q
SP_MCS : list of super-peers composing the MCS
including SP_Q (Can treat the query Q)
C_SP : community of SP_Q
SP_MCS = MCS(SP_Q)  // MCS including  SP
IF empty(SP_MCS) Then
   C_SP = Com(SP_Q)// Search for the community
                    of SP_Q
   SP_MCS = MCS(C_SP, k) // return the k MCS
        associated to the community of SP_Q
EndIF
AQ = {},   RQ = {}
For X ∈ SP_MCS Do
  For SP ∈ X Do
     Send(Q, SP)   // and other KSPi
     Local_Search(Q, SP)   // Search for peers of
                 // SP can treat Q: Local Search
     IF (( ∃ Son(Pk, SP)) ∧ (CAP(Pk)>ε)) Then
           // Pk can treat Q
      AQ  =  ∪  {SP@Pk}
```

Algorithm 2: k-Filter Calculate the Responses of Q using the MCS (Cont.)

```
      RQ = ∪  Treat(Q, Pk)  // Pk treat the
                // query Q et return the response RQ
     EndIF
   EndFor  // End treatment for MCS
EndFor  // End treatment for K MCS
Return AQ, RQ.
```

The function (MCS C_SP, k) returns the k MCS associated with the community C and are candidates for processing the query Q of super-peer SP. This choice may be more or less complex. For that we order the MCS of a community according to their size (number of super-peers that composed its), then take the first k. The algorithms 1-Filter and *-Filter can be reduced to the k-filter algorithm respectively using a single MCS or all MCS of the community c. The algorithm 2 selects only one Strategy, set of super-peers, and sends the queries, considering only its super-peers (belongs to minimal traversal), then any super-peer in the community, in order to use the CAP to choose relevant(s) pair(s) to respond to a given query, taking into account their ability to respond to the query Q.

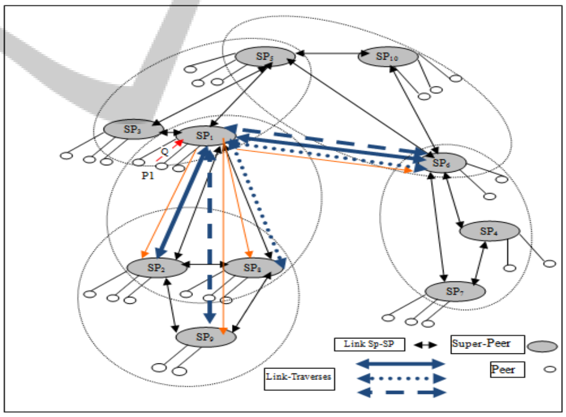

Figure 4: Routing queries using multiple sleepers.

Example (1-Filer): In this example, we use a single strategy {SP_1, SP_2, SP_6}. A query Q is sent by the super-peer SP_8, which in its turn, sends it to the super-peer SP_2 belongs to the traverse {SP_1, SP_2, SP_6}, then to all super-peers {SP_1, SP_6} belonging to the same traverse and therefore the query arrived to all communities of the P2P network. And consequently, the query is sent to all super-peers in a community (see Figure 4).

Example (3-Filter): In this example, we use three traverses for the super-peer SP_1 by example {{SP_1, SP_2, SP_6}, {SP_1, SP_6, SP_8}, {SP_1, SP_6, SP_9}}. A query Q is sent by the super-peer SP_1, which in its turn, sends it to all super-peers {SP_2, SP_6, SP_8, SP_9}

(see Figure 4), because in this super-peer SP$_1$, we have all the traverses that passed by this super-peer (such as routing table for queries.)

5 EXPERIMENTS AND EVALUATIONS

We describe the performance evaluation of our routing algorithm with a SimJava-based simulator. In our experimental study we compared the performance of our proposed system (Traverse) with the SenPeer (Faye et al., 2007).

In our experiments, we have accomplished various simulations respecting the following protocol:

- For each class of architectures defined by the number of peers NP and super-peers NSP, we generate the set of domains D and the query Q. As peer generates a query by using its schema, the number of queries is equal to the number of peers.
- It generates peers, super peers and schema
- The above parameters are then used to simulate the Baseline architecture and the Community architecture MCS.

This allows us to compare architectures using the same semantics on the same network and the same requests. We systematically measured, for each architecture, the average time to answer queries, the precision and recall. As our objective is to study the contribution of communities in the network SON, then we compared with the approach Baseline (Faye et al., 2007). For each pair of architecture, we measured the average response time (Figure 5) and accuracy (Figure 6) and recall (Figure 7). These simulations show a significant improvement in the results of our community-based approach compared to the Baseline approach.

The Figure 6 compares the accuracy of the community method over Baseline method. The Baseline method is based on the notion of capacity of a peer to answer a query. This capacity is based on the expertise of the peer and the query. Indeed, we consider that a peer has the capacity to process a query if the number of components of the query corresponding to the expertise elements of a peer. In this example, we consider a peer has the capacity to process a query if the expertise is at least two components of the query. Therefore it is possible that a peer with this method can be classified as having the ability to process the query without being able to really treat. This is a problem in the Baseline approach.

The Figure 6 shows, for a network of size less than 5000 peers, the accuracy of the MCS architecture (97%) compared to the architecture Baseline (87%). We clearly see the difference between the architecture MCS and the Baseline architecture.Finally, a recall of the Baseline approach is the lowest compared to recall of Community method because it is based primarily on research involving friends of super-peer that received the query. However, we can find other peers can process a given query without being friends with the super-peer that has received this query. The search space is reduced to the Baseline, while the extension of this space in the case of community-based approach allows them to increase their recall.

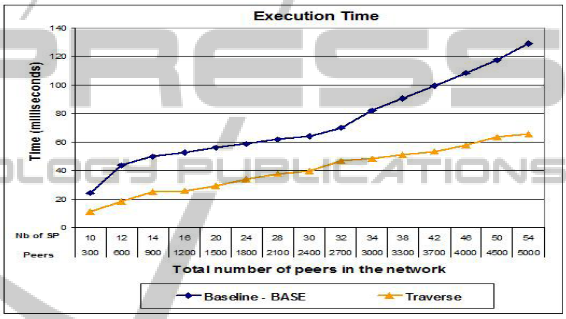

Figure 5: Time Evaluation: Baseline-MCS.

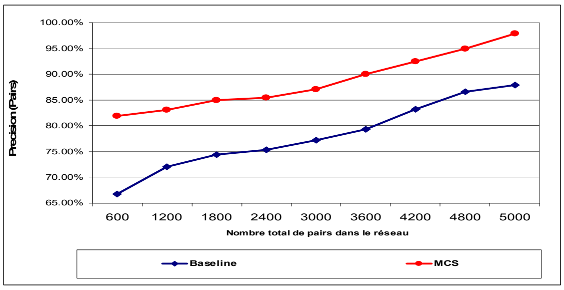

Figure 6: Precision Baseline – MCS.

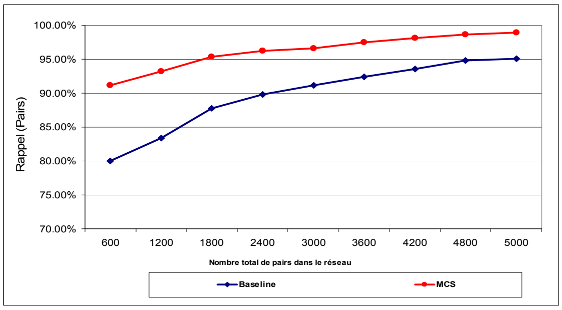

Figure 7: Recall Baseline – MCS.

Recall increases with the size of the network and reaches a percentage about 98% in the MCS architecture, 97% in the hybrid architecture (CK-Baseline) and 95% in the Baseline architecture.

We compared the simulation results of the tow approaches while showing their sensitivity to some adjustable parameters. It should be noted that these results are also influenced by the mapping between the super-peers. Indeed, this mapping is more important (dense) more queries are received by a super-peer are broadcasted to a larger number of super-peers, which automatically increases the time required to answer a query.

6 CONCLUSIONS

In this paper, we explored the contribution of grouping domain within the SON. We were placed in a specific context where each Peer has its own schema and is related to a SP. As a Background, a SP topology as a suitable topology for schema-based P2P networks was discussed, and how additional clustering in such network can be used for query routing among peer communities. We proposed an advanced method using hypergraph-based algorithm with minimum traversal to route a given query. The advantage of this model is the robustness in Queries routing and scalability issues in P2P Network One important area for improvement is performance.

An important problem remains unresolved for communities approach proposed which is the influence of the dynamics of groups and their characterizations on system performance. Indeed, it is possible to introduce mechanisms that allow a community to exclude a member (inactive Super-Peer) or to accept a new one (newcomers or those whose interest has evolved).

ACKNOWLEDGEMENTS

This work has been done as a part of the following projects: "Automatic information extraction form Arabic texts" supported by the CNRSL, "Extraction automatique d'information à partir des documents Arabes", supported by the UL, and "Arabic speech synthesis from text, with natural prosody, using linguistic and semantic analysis" supported buy the PCSL.

REFERENCES

Aberer, K., 2001. P-grid: A self-organizing access structure for p2p information systems. *In CoopIS*, pages 179-194.

Aberer, K., Cudre-Mauroux, P., Hauswirth, M. and Pelt, T. V., 2004. Gridvine: Building internet-scale semantic overlay networks. *In International Semantic Web Conference*, pages 107-121.

Akbarinia, R. and Martins, V., Data management in the appa p2p system, 2006, *In Int. Workshop on High-Performance Data Management in Grid Environments (HPDGRID)*.

Bhaduri, K., Wolf, Giannella, R. C. and Kargupta, H., 2008, Distributed Decision Tree Induction in Peer-to-Peer Systems. *In Statistical Analysis and Data Mining Journal*, volume 1, pages 85-103.

Castano, S. and Montanelli, S., 2005. Semantic self-formation of communities of peers. *In Proceedings of the ESWC Workshop on Ontologies in Peer-to-Peer Communities*, pages 137-151.

Cohen, E., Fiat, A. and Kaplan, H., 2003. Associative search in peer to peer networks: *Harnessing latent semantics*.

Crespo, A. and Garcia-Molina, H., Routing indices for peer-to-peer systems, 2002. *Distributed Computing Systems, International Conference on*, pages 0-23.

Cuenca-Acuna, F. M., Peery, C., Martin, R. P. and Nguyen, T. D., 2003. Planetp: Using gossiping to build content addressable peer-to-peer information sharing communities. *HPDC,* 236-249.

Datta, S., Giannella, C. and Kargupta, H., 2006. K-means clustering over a large, dynamic network. *In J. Ghosh, D. Lambert, D. B. Skillicorn, and J. Srivastava, editors, SDM. SIAM.*

Durand, N. and Cremilleux, B., 2002. ECCLAT: a New Approach of Clusters Discovery in Categorical Data. *In 22nd Int. Conf. on Knowledge Based Systems and Applied Artificial Intelligence (ES'02),* pages 177-190, Cambridge, UK.

Durand, N., Cremilleux, B. and Suzuki, E., 2006, Visualizing Transactional Data with Multiple Clusterings for Knowledge Discovery. *In Proc. 16th Symposium on Methodologies for Intelligent Systems (ISMIS'06)*, pages 47-57,.

Faye, D., Nachouki, G. and Valduriez, P., 2007. Semantic query routing in senpeer, a p2p data management system. *In NBiS*, pages 365-374,.

Hebert, C., Bretto, A. and Cremilleux, B., 2007. A data mining formalization to improve hypergraph transversal computation. *Fundamenta Informatica, IOS Press*, 80(4), pages 415-433.

Ismail, A., Quafafou, M., Durand, N., Nachouki, G., Hajjar, M., 2010. Queries Mining for Efficient Routing in P2P Communities, International Journal of Database Management Systems (IJDMS), vol. 2, No. 1, pages 9-28.

Tong, X., Zhang, D. and Yang, Z., 2005. Efficient content location based on interest-cluster in peer-to-peer system. *E-Business Engineering, IEEE International Conference* on, pages 324-331.

Ratnasamy, S., Francis, P., Handley, M., Karp, R. and Schenker, S., 2001, A scalable content-addressable network. *In SIGCOMM '01*, volume 31, pages 161-172.