Queries Routing In Super-Peer-Based System: Simulation and Evaluation

Anis ISMAIL & Mohamed QUAFAFOU
LSIS, Domaine Universitaire de Saint-Jérôme,
Avenue Escadrille Normandie-Niemen
13397 MARSEILLE CEDEX 20
Email: {anis.ismail, mohamed.quafafou}@univmed.fr

Gilles Nachouki
LINA Laboratory
2 rue de la Houssinière
44322 Nantes cedex 03
Gilles.Nachouki@univnantes.Fr

Mohammad HAJJAR
Université Libanaise, Institut Universitaire de Technologie
Saida – LIBAN
m_hajjar@ul.edu.lb

Abstract— **Peer-to-peer (P2P) computing is currently attracting enormous attention. P2P systems have emerged as a popular way to share huge volumes of data. In such systems each peer is a database management system in itself, ex-posing its own schema. A fundamental problem that confronts peer-to-peer applications is the efficient location of the node that stores a desired data item. In such settings, the main objective is the efficient search across peer databases by processing each incoming query without overly consuming bandwidth. In this paper, we propose an architecture based on (super-)peers, and we focus on query routing. Our approach considers that (super-)Peers having similar interests are grouped together for an efficient query routing method. In such groups, called Knowledge-Super-Peers (KSP), super-peers submit queries that are often processed by members of this group. A KSP is a specific super-peer which contains knowledge about: 1. its super-peers and 2. The others super-peers. Knowledge is extracted by using data mining techniques (e.g. decision tree algorithms) starting from queries of peers that transit on the network. The advantage of this distributed knowledge is that, it avoids to making semantic mapping, between heterogeneous data sources owned by (super-)peers, each time the system decides to route query to other (super-)peers. The set of KSP improves the robustness in queries routing mechanism and scalability in P2P Network. Compared with a baseline approach, our proposal shows a better performance using a new simulator with respect to important criteria such as response time, precision and recall.**

Index Terms—**Peer-to-peer, Query Routing, Knowledge-Super-Peers, Data Mining, Scalability.**

I. INTRODUCTION

Peer-to-peer (P2P) systems have recently become a popular medium through which huge amounts of data is shared. Because P2P systems distribute the main costs of sharing data – disk space for storing files and bandwidth for transferring them – across the peers in the network, they have been able to scale without the need for powerful and expensive servers. The key to use a data-sharing P2P system, and that is the one of the most challenging design aspects, is efficient techniques for search, route queries and retrieval of data. The major problem in such networks is query routing, i.e. deciding to which other (super-)peers the query has to be sent for higher efficiency and effectiveness.

However, such systems that broadcast all queries to all peers suffer from limited efficiency and scalability, and are difficult to locate files which result also in much network traffic and low recall/precision. In hybrid P2P systems [1][2], composed of (super-)peers, when a peer submits a query, this peer becomes the source of this query. Then the query is transmitted to its super-peer (SP). The routing policy in use determines quickly the relevant neighbors (i.e. SP), based on semantic mappings between schemas of (super-)peers, and to which neighbors, the query is sent. When a SP receives a query, it will process the query over its local collection of data sources of different peers and sends the query to the relevant neighbors (SP) for processing [24]. If any results are found, the SP will send a single response message back to the query source. Another important aspect of the user experience is how long the user must wait for the results to arrive. This is due to a large part of the mediation process which remains difficult to realize in such a context when the number of (super-)peers increases. Response times tend to be slow in hybrid P2P networks, since the query travel through several SP in the network and because the SP is forced to look for connections (i.e. mappings) in order to route the query.

Satisfaction time is simply the time that has elapsed from when the query is first submitted by the user, to the other users that contain the most relevant answers in a fast and efficient way, until the user receives the overall

doi:10.4304/jetwi.3.3.206-216

results. This also is the main challenge of information retrieval in Peer-to-Peer networks [12].

In this paper, we present an approach for efficient queries routing. The important advantage of this approach is scalability. Our system is designed to efficiently support content-based searching. Our main goal is to reduce the processing of queries at the SP level to predict others relevant SP in order to receive and process such queries. Our proposed method focus on how the query is routed to relevant Peers with minimum query processing in order to improve answering time of the queries.

Our approach consists of grouping together (super-)Peers that have similar themes for an efficient query routing method. Each obtained group, called Knowledge Super-Peers (KSP), contains Domains, composed of super-peers (the responsible of Domains) and their corresponding peers (the members). These peers submitted queries that are often processed by members of this group (after grouping). Each KSP operates with an index that, obtained by applying decision tree algorithms, keeps track of where contents concerning a query are located: when a KSP receives a query from a Super-Peer (in his group), it consults directly its index (without making any mappings) in order to determine: 1. in his group all super-peers (or Domains) that are able to answer this query and 2. in other groups (i.e. other KSP) all super-peers which are relevant to this query. In this paper, we do not care how we get the different groups of SP but we focus only on the Super-Peer based routing protocol of users's queries. We have also implemented a new simulator to evaluate our approach.

The following section recalls briefly principal concepts of P2P networks and shows the context of our work. Section 3 presents the baseline algorithm of queries routing in hybrid P2P systems. Section 4, introduces the Knowledge super-peer (KSP) network. Section 5 presents the semantic routing of queries algorithm. Section 6 presents our simulator. Section 7 presents Experiments and Evaluations. Section 8 presents a related work. In Section 9, we present the conclusion and future works.

II. Background

A. Basic notions

In a computer network, a Peer may act as a client or as a server. A P2P is a set of autonomous and self-organized peers (P), connected together through a computer network. The purpose of a P2P network is the sharing of resources (files, databases) distributed on peers by avoiding the appearance of a peer as a central server in this network. We note: P2P = (P, U), P is the set of peers and U represents links (overlay connections) between two peers P_i and P_j, $U \subseteq P \times P$.

The hybrid P2P ($P2P^h$) (See Figure 1) network that we consider in this paper includes sets of peers (P) and super-peers (SP). We note : $P2P^h = (P \cup SP, K)$, where P is the set of peers, SP is the set of super-peers and K is the set of overlay links expressed under the format of pairs : (P_i, SP_j) or (SP_j ,SP_k) which respectively link a Peer P_i to a Super-Peer SP_j or a Super-Peer SP_j to one or several super-peers SP_k.

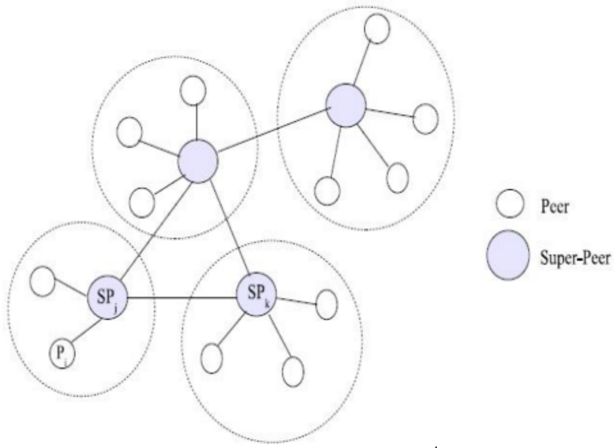

Figure 1. Hybrid network ($P2P^h$).

A PDMS (Peer Data Management System) combines P2P systems and databases systems. The PDMS that we are considering is a hybrid scale system $P2P^h$. Each peer is supposed to hold a database (or an XML document, etc.) with a data schema. Each Super-Peer provides a theme (a semantic domain, a subject, or an idea) representing special interest to a group of peers.

We note R the set of relations reduced in this paper the PDMS={PS $\cup$ SP,T, D , K} where PS represents all the peers of the network with their data schemas S={S_1, …., S_p}. A peer is connected to the network with only one data schema. K is the set of overlay links between (super-)peers. Each peer P $\in$ PS is doted of a Data Management System (denoted DMS) able to manage their data.

T={T_1,…., T_k} represents the interest themes published by super-peers SP through the network. In our case, each super-peer publishes only one theme and peers expresses that are interested by one or several theme(s) in T. The themes are not disjoints: two super-peers can publish the same concepts or roles with distinct structures and/or don't use the same vocabulary. D = {D_1, …., D_k} describes the themes in the set of T: D_j describes the theme T_j specifying the set of concepts and their relationships (see figure 2).

B. Expertise, Mapping and Domains

To facilitate the reconciliation, between the data schema of the Peer and the theme described by a Super-Peer, two measures were taken: 1. the expertise of a Peer is expressed with the language of its Super-Peer (i.e. concept, role and IsA); 2. The expertise of a Peer is expressed under the format of couple of elements, satisfying the following condition:

$$\mathrm{EXP}(P_i) = \{\theta(s_i; s_j) \in SP \mid (s_i; s_j) \wedge \theta \in R\}$$

We introduce the two concepts, Semantic Intra-domain and Semantic Inter-domain. A Semantic Intra- domain is an interest domain in which mappings between peers (members of this domain) and the Super-Peer (responsible of this domain) are established. A Semantic Inter- domain is a set of semantic Intra-domain in which mappings between Super-peers of these domains are established.

We note Semantic Intra-domain $(SI_a^j C)$ and Semantic Inter-domain $(SI_a^j C)$ number j as follows:

$$SI_a^j C = (\mathrm{PS} \cup \mathrm{SP}_{\mathrm{Tj,Dj}}, \mathrm{EXP}(\mathrm{P_s}), \mathrm{K}^j ; \mathrm{RSC}^j) \quad (1)$$

$$SI_a^j C = (SI_a^j C, RSI^{j-1}, \ldots, RSI^{j-k}) \quad (2)$$

where k $\neq$ j, $P_s \subseteq P$ is a subset of peers having the same center of interest T_j, EXP (P_S) is the set of expertise of peers interested by this theme and joined to this domain, $SP_{Tj,Dj}$ (belong to SP) is the Super-Peer responsible of the domain j which are joined by peers (i.e. a Peer of a domain may request to join several domains if the user thinks that his theme of interest is in the intersection of several domains), D_j represents the description of the theme T_j provided by the Super-Peer. $K_j \subseteq K$ is the set of overlay links between the super-peer $SP_{Tj,Dj}$ and the peers connected to it union the set of overlay links between $SP_{Tj,Dj}$ and Super-Peers $SP_{Tk,Dk}$, k$\neq$j, RSC^j is the semantic Intra- domain between the super-peer $SP_{Tj,Dj}$ and the peers inside this domain. $RSI^{j,k}$ is the semantic Inter-domain concerning the links found between the description of the theme D_j of the Super-peer $SP_{Tj,Dj}$, with the description D_k of each super-peer $SP_{Tk,Dk}$, k$\neq$j). Finally, we introduce a Semantic Overlay Network (SON) represented by the union of all the semantic networks of intra-domains and inter-domains. A SON is noted as follows:

$$SON = \cup_{j=1}^{|T|} (SI_e^j C) \quad (3)$$

Where T represents the total number of super-peers in the network. Next section presents the query routing algorithm (our baseline approach).

III. SEMANTIC QUERIES ROUTING - BASELINE

A. Network Configuration

A new Peer P_j advertises its expertise by sending, to its Super-Peer, a domain advertisement $\mathrm{DA}_j = (\mathrm{PID}; E_{XP}^j, \mathrm{T}_j; \varepsilon_{\mathrm{acc}}; \mathrm{TTL})$ containing the Peer ID denoted PID, the suggested expertise E_{XP}^j , the topic area of interest T_j , the minimum semantic similarity value (ε_{acc}) required to establish semantic mapping between the suggested expertise E_{XP}^j and the theme of its Super-Peer. When receiving an expertise E_{XP}^j , a Super-Peer SP_a invokes the semantic matching process to find mappings between its suggested schema and the received expertise.

B. Baseline approach

A Peer submits its query on its local data schema. This query is sent to its Super-Peer responsible for the domain (see Figure 2).

The Super-Peer in its turn suggests, based on the index obtained by the process of mediation (first level), the peers of his domain or the other super-peers 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 the expertise of peers (of the same domain) or the description of themes of other Super-peers. In its turn, a super-peer, from the nearby domain, having received this request, researches among peers (of his domain) that 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 scaled due to the mappings at both levels. The following sections describe our approach using Data mining.

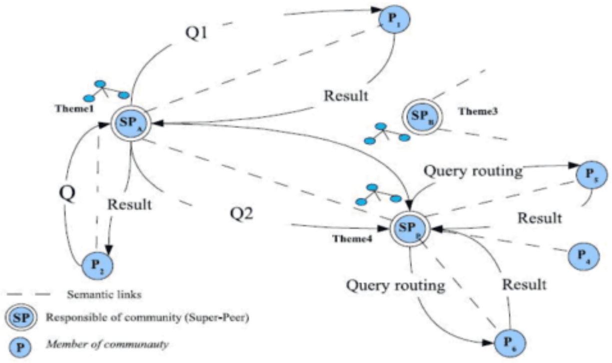

Figure 2. Network configuration and query routing (baseline).

IV. KNOWLEDGE-SUPER-PEER

A Knowledge-Super-Peer (KSP) network is a semantic sub-network of Overlay Network (SON). The KSP number j is defined as follows:

$$\mathrm{KSP}^j = \cup_{l=1}^{|M|} (SI_e^l C) \qquad |M| \le |T| \quad (4)$$

Where M is the number of Super-Peer in KSPj and |M|$\le$|T| (total number of super-peers). $SI_e^l C$ is the Semantic Inter- domain of the super-peer number l. Two fundamental properties are derived from KSP:

$$\mathrm{KSP}^i \cup \mathrm{KSP}^j = SON, \mathrm{i} \neq \mathrm{j} \quad (5)$$

A Knowledge-Super-Peer is represented physically with a specific Peer. This Peer, representing the Knowledge-Super-Peer number j, is noted as follows:

$$\mathrm{KSP}^j = (\mathrm{PS} \cup \mathrm{SP}_{\mathrm{TJ,DJ}}, \mathrm{EXP}(\mathrm{P_s}), \mathrm{K}^j, \mathrm{RSC}^J, \mathrm{RSI}^J, \mathrm{IND}^j)$$

where $P_S \subseteq P$ is a subset of peers having very close center of interests denoted T J = {T1,…, Ts}, EXP (P_S) is the set of expertise of peers interested by at least one of themes in T J, $SP_{T J, DJ}$ (belong to SP) is the set of super-peers responsible of domains which have very close domain interests, DJ = {D1, …, Ds} represents the description of themes in T J (DJ describes TJ). Kj $\subseteq$K is the set of overlay links between each super-peer $SP_{Tj, Dj}$ $\in$ $SP_{T J, DJ}$ and 1. The peers connected to it (within its domain); 2. The other super-peers; 3. The Knowledge-Super-Peer KSP^j itself. RSC^J is the set of semantic Intra-domain of the super-peers $\in$ $SP_{TJ, DJ}$. RSI^J is the set of semantic Inter-domain for each super-peer in $SP_{TJ, DJ}$. IND^j is the index obtained using a decision tree algorithm to identify directly the most relevant (super-)peers, without going through mappings, to provide good results when a query is submitted by a peer.

Our proposed System (See Figure 3) is an hybrid P2P system based on an organization of peers around super-peers according to their proposed themes, where super-peers are connected to a Knowledge-super-peer (KSP). A KSP is the engine that specifies the super-peers having peers which may have relevant data to answer queries with minimum query tasks and, by consequence, improve answering time of the queries. The super-peer architecture allows the heterogeneity of peers by assigning more responsibility to peers able to assume them.

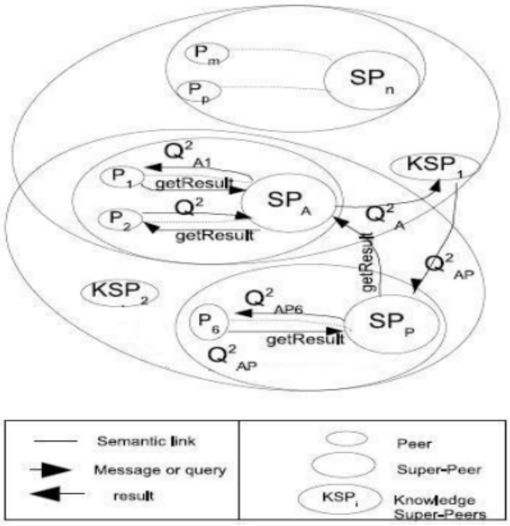

Figure 3. Network configuration and query routing (KSP approach).

Therefore, certain peers, called Knowledge super-peers, have an additional computing power and greater bandwidth, resources and performing administrative tasks. They are responsible of routing queries to relevant super-peers, allowing not only to reduce efforts of compilation of queries but also to prevent the spread of queries in the network. In each domain, there is a super-peer connected to a Knowledge super-peer where we have an index to identify super-peers that are most relevant to provide good results of queries.

The building block (KSP) of the current P2P systems in the architecture (Distributed Knowledge - DK) is the notion of a super-peer-group, or a number of nodes (super-peer) that participate with each other for a common purpose to minimize the load in the KSP. Example: In this example we explain the query routing using KSP (Figure 3). A Peer P_2 sends a query Q^2 to his SP (SP_A) that in its turn sends this query to it KSP that belong to it and also to the peers of its domain that are able to answer this query. This KSP analyzes the query to find the other SP using a decision tree to send this query. Finally, the results will be sent to P_2.

V. SEMANTIC QUERY ROUTING ALGORITHM

Our algorithm of semantic query routing is composed of two stages: the semantic routing algorithm (Algorithm 1) of the baseline approach exploits the expertise of (super-)Peers and the two levels of mappings in order to forward a query q to only the relevant Super-Peers. Each Super-Peer in its turn forwards this query to the relevant Peers of its domain.

Algorithm1: Baseline algorithm: BL(Q,SP)
Inprut: Q: Query
SP: Super-Peer of P
Output: SR_Q : Set of answers of Q
Variables: PSet: Set of Peers
NP: Neighbors of SP (Set of Super-peers)
$SR_Q = \phi$
Pset = $Capacity_{CM^{SP/SP}}(Q) > \varepsilon_{acc}$
repeat
SP_Q = get(s $\in$ PSet);
Remove SP_Q from Pset;
$SR_Q = SR_Q \bigcup Query(SP_Q)$;
Until (PSet= ϕ)
repeat
$SP_Q = Capacity_{CM^{SP/SP}}(Q) > \varepsilon_{acc}$
Remove SP_Q from NP;
$SR_Q = SR_Q \bigcup BL(Q,SP_Q)$;
Until(PSet= ϕ)
Return(SR_Q);

The followings sub-steps are necessary in order to process the query: 1. Extract the subject of this query; 2. select, by the super-peer, the most relevant peers for the query and the other super-peers (by matching the subject of the query to the set of expertise of peers or to the themes of super-peers). The selection is based on a function that measures the capacity of a peer or a super-peer in answering a given query; 3. Once the set of relevant (super-)peers has been identified, the super-peer sends the query to those promising peers or super-peers closed to them by using their ID, IP addresses and the underlying physical network. The advantage of this step is that it permits us, for the second step, to collect information about the queries received by super-peers and the relevant super(-peers) selected in order to process it. The second algorithm exploits the Knowledge-super-peers (KSP) network.

This algorithm (algorithm 2) is very useful when the performance of the system is low.

Algorithm2: Knowledge based algorithm KB(Q,SP)
Input: Q: Query
SP: Super-Peer of P
Output: SR_Q : Set of answers of Q
Variables: T_{SP}: decision Tree of SP
NP: Neighbors of SP (Set of Super-peers)
$SR_Q = \phi$
Pset = Select(p $\in$ SP)
repeat
SP_Q = get(s $\in$ PSet);
Remove SP_Q from Pset;
$SR_Q = SR_Q \bigcup Query(SP_Q)$;
Until (PSet= ϕ)
$SP_Q = T_{SP}(Q)$;
$SR_Q = SR_Q \bigcup Query(SP_Q)$;
Return(SR_Q);

This step runs in three stages: 1. the super-peer sends the query directly to its Knowledge super-peer; 2. the

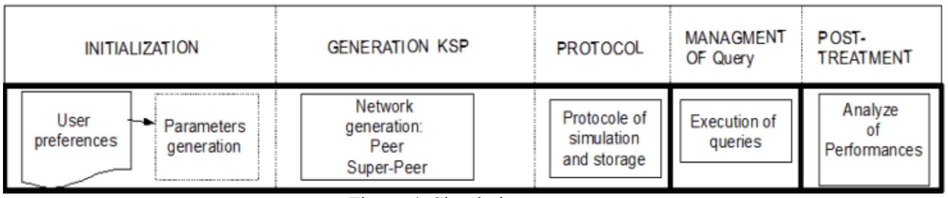

Figure 4. Simulation process

Knowledge super-peer identifies (without making mapping) the relevant KSP of this query and their super-peers by consulting its index IND (obtained by applying decision tree algorithms); 3. Each selected super-peer sends the query to the relevant Peers; 4. The final result of selected peers is returned.

VI. SIMULATOR ARCHITECTURE

The simulation is a technique to model the real world [27]. It can represent the operation of a system consisting of various activity centers; it reveals the characteristics of them and the interactions between them. It also describes the movement of the various subjects treated by these processes. Finally, it permits to observe the behavior of the system as a whole and its evolution over time. The discrete event simulation can help understand the behavior of the system. Several research projects such as Freenet [28] and Anthill [29] have used simulation in order to show their performance. The discrete event simulation allows observing the behavior of the system. The model has a state described by variables that define completely the characteristics of the system [30].

There are several peer-to-peer simulators available: P2PSim [31] is a discrete event simulator for structured overlay networks written in C++. It comes with seven peer-to-peer protocols implemented including the more recent protocols Koorde [34] and Kademlia [35].

OverlayWeaver [36] is a peer-to-peer overlay construction toolkit written in Java which can be used for easy development and testing of new overlay protocols and applications.

The state model is often encapsulated in a set of entities (objects in object-oriented programming). The discrete event changes the system state that occurs at different points in time (as opposed to the continuous change of states). Events may trigger new events. Statistical variables then define performance measures relevant to the user.

In this section, we present our P2P network simulator domain-based semantics. The simulation process that we present in Figure 4 consists of five main stages, each supporting a set of generic functions:

1. **Initialization**: The initialization phase permits to acquire the user preferences. These preferences mainly concern the number of peers, super peers, the various fields (super-) peers and the choice of strategy (semantic-oriented or knowledge-based ties) to be used during the Management queries. Based on user preferences, a set of parameters common, to all other strategies, is generated. These parameters are mostly the identification of areas of expertise and the generation of super-peers.

2. **Generation KSP**: The generation phase KSP network is an important step in the process of simulating semantics P2P network. This phase permits to construct and simulate KSP networks accordance architectures presented in this paper.

3. **Protocol**: the protocol allows specifying some basic rules necessary for the proper functioning of the simulator. On the stage "Managing Queries", the simulator must know which method to adopt to send the queries: for example, a first method is to generate a query in pairs; queries are generated and sent along with the super- peers for processing. Another method is to generate multiple requests (a number of query i randomly chosen between 1 and N) per pair. Regarding the management of the network, the simulator needs to have information on domain-groups: they can be dynamic. For example, on one hand a super-peer in a domain-group may transfer to another domain-group if it does anything (knowledge) to this domain-group and on other hand, a super-peer may leave the network completely. Regarding the management of knowledge in a domain-group, we can distinguish cases where knowledge at the domain-group level can be static or dynamic. Knowledge dynamics are updated periodically by the relevant domain-group.

4. **Management of Query**: This phase involves generating a plan for routing queries. The generated plan is built by one of the strategies described in this paper: semantics or domain-groups. The KSP approach is a hybrid approach since it is based on two strategies: knowledge-oriented or minimal ties (to search for relevant super-peers can answer a query) and semantic (search within each Super-Pair relevant peer that can respond to this query).

5. **Post-treatment**: This phase involves defining the types of expected results and analyzes the performance of each simulation performed.

A. Semantic Aproach

In the semantic approach (Figure 5), several parameters are needed to build the semantic SON. Among these parameters, we include the number of peers and super-peers that make up our network areas of (Super-) peers; different thresholds: 1. A level of correspondence (mappings) deemed acceptable by the (Super-) peers; 2. An acceptable threshold for establishing trust between two super-peers and 3. The ability of a (Super-)peer to process a request. These parameters are common for different strategies.

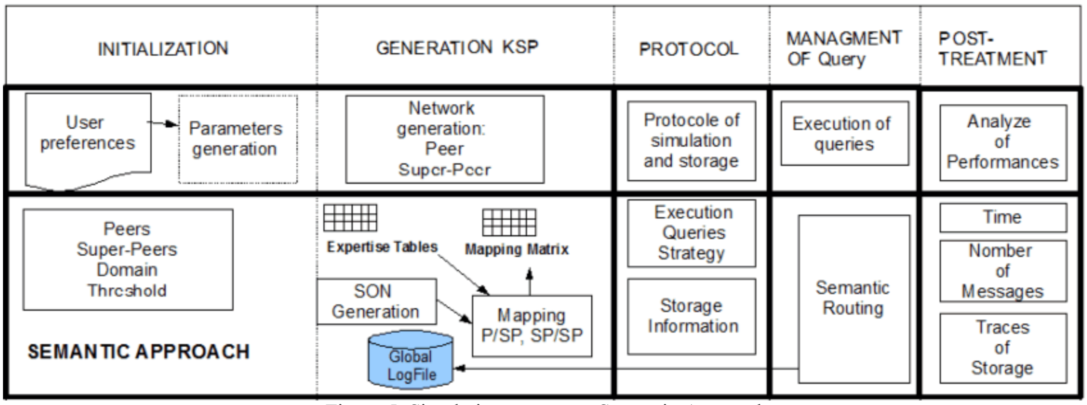

Figure 5. Simulation process – Semantic Approach

The algorithm 3 initializes the system with the generation of areas and expertise of the super-peers:

Algorithm 3: Generation of domain parameters

```
    Begin
    Entry:
      NP: number of peers
      NSP: number of super-peers

    Released:
      List domain D: list of generated Domains
      List_expertise_E: List of expertise of the super-peer
    Begin
    For i = 1 to NSP Do // generate fields of super-peers
1:      List_domain_D= Liste_domain_D ∪
        Generate_domain_D = (di)
    EndFor
    For i = 1 to NSP Do //generation of expertise super-peer
2:                        SP
        List_expertise_E = List_expertise_E ∪
        Generate _domain_E(SPi)
    EndFor
    Return List_domain_D, List_expertise_E
    End.
```

The size of the network being defined by the number of peers NP and super-peers NSP that are given by the user. For each super-peer I, we generate using the function generate_domain a label di which is the name of the domain represented by the super-peer i (step 1 of algorithm 3). This generation respects the following condition: two super-peers can not be assigned to the same domain. Then it generates, in step 2, the expertise of each super-peer represented as a set of couple (x, y). We note χ (X) the expertise of the super-peer X.

To generate the SON networks, we start building the correspondences (mapping) between the super-peers. Then, we generate peers, their expertise and we implement the connections between peers/super-peers. It begins by calculating the correspondence (mapping) between the semantic super-peers (algorithm 4). For this, we represent the expertise of super-peer by an expertise table (ExpTabSP) of super-peers. To simulate this calculation, a super-peer selects randomly a number of super-peers of the network to consider them as friends, and then duplicate some elements of its expertise in the expertise of his friends. The number of duplicate elements has to be selected in order to ensure the existence of mapping between a super-peer and his friends.

Algorithm 4: Generation of SON Network (SP/SP)

```
    Pre-condition: SPi is a new super-peer in the network
    Begin
     Entry:
       ExpTabSP table d'expertise SP
       CorMatSPSP: Matrix correlation SP/SP
     Output:
       TFI: table of friends of the super-peer SPi (initially
            empty)
       CorMatSPSPi: Changing the correlation matrix
    TFI = Select_ friend (SPi) // selects SPi Friends
    For each super-peer SPj ∈ TFi Do
      T = select expertise (ExpTabSPi) // selection elements exp. SPi
1:    Send (SPi, T, SPj) / / Send selected elements to SPj
2:    Addition (SPj ExpTabSPj, T) // SPj ExpTabSPj addition
                                     to the elements of T

      Addition (SPj, TFj, T, SPi) // SPj has a new friend SPi
3:
      Update (SPi, CorMatSP) // update mapping SPi
4:    Update (SPj, CorMatSP) // update mapping SPj
    EndFor
    End
```

At this level, the SON network is built; it remains to clarify the evolution of its architecture based on the dynamics of peers and super-peers.

Trust between two super-peers depends on the number of semantic links connecting them. The trust is useful where a super-peer SP leaves the network: peers attached to SP will then be attached to the super-peer with the highest degree of trust with SP.

Algorithm 5: Generation of SON Network (P/SP)

```
    Begin
     Entry:
       NP: number of peers
       MIN: minimum size of the expertise of a peer
    Output:
1:    ExpTab table of expertise of P
      CorMatSPP: correspondence Matrix of super-peer/peer

2:  For i = 1 to NP do // generate expertise of peers P
       List expertise=generate expertise (Pi, SPi, MIN)
      ExpTab = store (List expertise)
3:    Storage(ExpTab, SPi)
    EndFor
    Create nodes (SP) // creation of SON Network
    Create nodes (P)

    CorMatSPP= Create_Correspondance = (P, SP) //create link
    peer/super-peer
    End.
```

Algorithm 6: Query routing, Generation of global LogFil

```
     Pre-condition: The Queries are in the parameters file
     identifier strategy (ids = 1)
     Begin
     Input:
      ExpTabSP: Table of expertise associated with the super-peer SP
      ExpTabP: Table of expertise associated with the peer P
      Threshold: threshold acceptable

     At time t: ∀ P of the network has SP as super-peer Do
1 :   send(P, Q, SP) // P sends its Query Q to SP

     Perform local search
2 :   List P = search(SP, ExpTabP, Q) //search pertinent
                                           peers
3 :   While Pk ∈ List P Do
       Send(SP, Q, Pk) // Send Q to Pk
      EndWhile
4 :  Perform global search
     A = Friends(SP) // A all the super-peer friends of SP
5 :   While SPk ∈ A Do
6 :   List_SP = search(SP, ExpTabSP, Q) // Search pertinents SP for
7 :  Q
          While SPk ∈ List SP Do // for all SP that can
8 :                                   process the Queries
             Send(SP, Q, SPk) // Send Q to SPk
                         // SPK performs a local search
              List P = Search(SPk, ExpTabP, Q)
              //Search pertinents P
                  While Pi ∈ List P Do
                    Send(SPk, Q, Pj) // Send Q to Pk
                   Endwhile
           Endwhile
      Endwhile
     End
```

We consider that the queries are expressed in the simulator in the form of elements of expertise: $\bigwedge_{i=1}^{n}(p_i \cdot q_i)$ where $p_i.q_i$ is easily compared with the components of expertise of peers and super peers. It is considered that the rewriting query of the user under the form of expertise elements is not part of the simulator, but it is a task delegated to the mediator.

The generation of applications is ensured by peers. In fact, each peer P can generate a query by selecting elements of expertise that become components of the query Q. We say that a peer P is relevant to the query Q if the expertise of P contains at least a fraction of the components of Q. This is determined using the ability of a peer P to resolve a query Q.

So each peer generates a number N of queries that are derived from its expertise. After this phase generation of query, peers send their queries to their super-peers. Algorithm 6 shows in detail the stage for routing queries in the context of the semantic approach. In fact, step 1 shows that all peers send their queries to their super-peer at time t. The super-peer that receives the query performs a local search (step 2) by only considering one pair that belongs to the domain it represents. Then, the super-peer sends the query to his friends that can respond to the query for global search (step 4).

All queries exchanged within the network are stored in a file global LogFile. Thus, for a query Q, the file LogFile contains the following information: the identifier of the peer (P), which submitted the application, its super-peer (SP), the query (Q) itself and the super-peer which responded favorably to this request.

B. Aproach KSP

In this section we present an SON-KSP network based on knowledge. First, we begin by simulating domain-groups oriented knowledge (Figure 6). At this level, domain-groups are built above the previously established semantic layer, based on reliance by a member from one domain-group to another domain-group member. Indeed, a super-peer (referred to as a domain member) is free to join a domain-group if at least one member of this domain-group has given him confidence.

In this knowledge-oriented approach and to initialize the system, the user must give the acceptable threshold for establishing trust between the super-peers and must decide the dynamics of knowledge within domain-groups (refresh knowledge) then begins the generation of SON-KSP network. At this level, we extend the SON network to SON-KSP. A domain-group is characterized by the knowledge that bears on its super-peer as well as super-peers in neighboring domain-groups. According to user preferences at this level we build one or more domain-groups. Building a central domain-group is directly from the global LogFile previously built (strategy based on the semantics). The construction of several domain-groups is mainly based on the notion of trust referred to in the

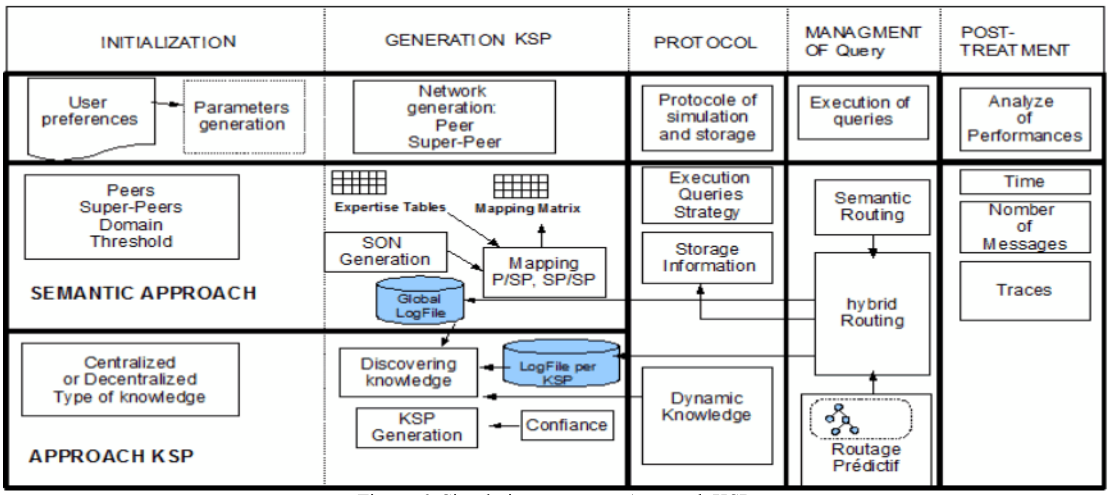

Figure 6. Simulation process – Approach KSP

preceding section. Indeed, the SON network is built from a layer dedicated to peers and another juxtaposed containing super-peers, and above a third layer was built domain-groups (SON-KSP) where each node is a domain-group.

The knowledge-oriented approach combines knowledge of each domain-group it owns (SON-KSP). Before extracting the knowledge of domain-group, we need to involve each domain-group its log file containing all the queries processed by one of its members (super-peer). The data contained in this file will be analyzed by domain-group using a tool data mining to extract knowledge. The role of knowledge in this context would be to predict the super-peers that may treat a given query. We express this knowledge in the form of a decision tree.

In practice, we build knowledge of different obtained domain-groups. We used the J48 algorithm implemented by WEKA [25] and its inference methods to find the probable super-peers to treat a given query. The traces of the simulation are stored in different files to support post-processing methods in order to analyze and compare the results of several simulations.

VII. EXPERIMENTS AND EVALUATIONS

Decision trees represent a supervised approach of classification. We have used Weka [25] in our experiments. The most important part of the entire data mining process is preparing the input for data mining investigation.

Our P2P database contains data from more than 300 peers with 10 super-peers (data contains the keywords (composant W1...W4) search of queries (part of expertise of peers (k.f, p.i, f.p, g.h, ...) and their answers (relevant peers with their super-peer)), after simulation the baseline Architecture and apply the data mining rules (Extraction and filtering data) to obtain the ARFF format that is input data to be injected in Weka to obtain the decision tree. Decision trees are often used in classification and prediction. It is a simple and powerful way of knowledge representation. The models produced by decision trees are represented in the form of tree structures. A component of query indicates the class of the examples. The instances are classified by sorting them down the tree from the first component of the query to other component of the query.

- The most popular measure for the effectiveness of our systems is the precision and recall.

Decision trees represent a supervised approach of classification. Weka uses the J48 algorithm, which is Weka's implementation of C4.5 Decision tree algorithm. J48 is actually a slight improved the latest version of C4.5.

We describe the performance evaluation of our routing algorithm with a SimJava-based simulator [6]. All experiments were run on a machine Core 2 Duo 1.83GHZ with 4 GB RAM, 250 GB Hard disk and Windows Vista operating system. Evaluating the performance of P2P network is an important part in understanding how useful it can be in the real world. As with all P2P applications, the first question is whether P2P is scalable. Our systems were evaluated with different set of parameters i.e. number of Peers, number of Super-Peer etc. Evaluation results were quite encouraging.

There are many dimensions in which scalability can be evaluated: one important metric is the time it takes the Answer of a given query. We run simulations on P2P network in three different sizes. Each peer sends a query to its SP that sends the query to a KSP in order to precise which Super-peer(s) can answer the given query (this in the architecture-DK).

- First one, we modified the number of Peers (300, 600,..., 5000 Peers) and Super-peers (10,12 ,14, 16, 20,..., 54) in both Architectures to measure the execution time.

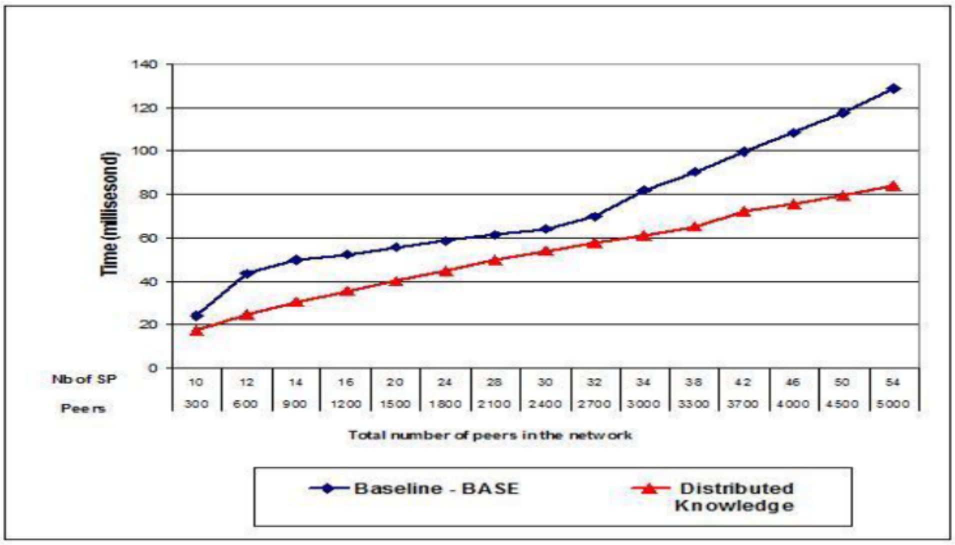

Figure 7. Response time

$$precision = \frac{\#\, of\ relevant\ \text{Responses}\ retrieved}{total\ \#\, of\ retrieved\ \text{Responses}}$$

$$recall = \frac{\#\, of\ relevant\ \text{Responses}\ retrieved}{total\ \#\, of\ relevant\ \text{Responses}}$$

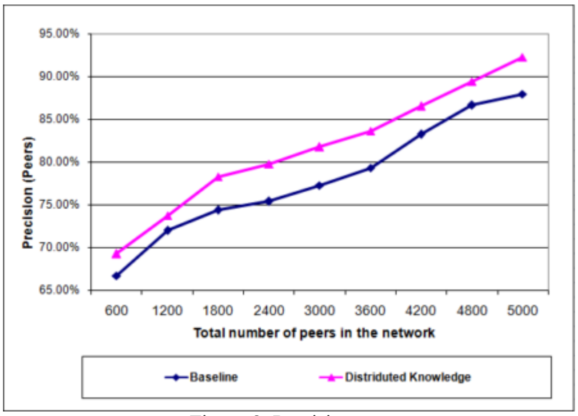

Figure 8. Precision rate

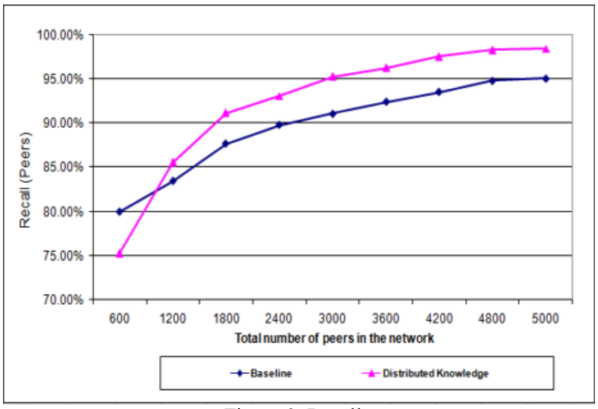

Figure 9. Recall rate

The Graphs shown in figures 7, 8 and 9 are the results of our simulations. They demonstrate the performance of using the Knowledge Super-Peer with a decision tree for routing Queries to the relevant P2P domains (SP). In the first observation, the difference in the execution times between 300 and 600 peers in the DK architecture is small (See Figure 7). The execution time was measured as the repository size increased.

Measurements, shown in Figure 7, show that the time increased in the DK architecture less than the baseline architecture. In the DK architecture at 5000 Peers, the response time decreases about 35 % of the baseline architecture, this is due to the presence of prediction mechanism in DK architecture. Measurements in Figure 7 show that the execution time decreased where the number of peers and super-peers (domains) increased. This means how much our DK architecture is scalable. Measurements in Figure 8 have shown the precision of the DK architecture compared to the Baseline architecture. In the DK architecture, we observe that the precision will increase comparing to the baseline architecture due to the knowledge of all domains including in the KSP. However in the baseline architecture, we have correspondence between the neighborhood domains. In addition, This experiment was designed to measure the accuracy of data (since precision is almost not affected completely by the network size) which is the recall (See Figure 9).The recall increases with the size of the network and reaches a percentage of almost 95 % in the DK architecture; Whereas in the baseline architecture, it reaches about 91% because the baseline reduced the research space however the DK architecture increased this space research area. Finally, our Prototype raises some interesting performance issues while grouping P2P domains (P2P). We perform experiments to demonstrate how the presence of grouping domains affects the performance and, in addition, to illustrate how grouping domains can improve the scalability of the overall system.

VIII. RELATED WORK

P2P networks are quickly emerging as large-scale systems for information sharing. Through networks such as Kazaa, e-Mule, BitTorrents, consumers can readily share vast amounts of information. While initial consumer interest in P2P networks was focused on the value of the data, more recent research such as P2P web community formation argues that the consumers will greatly benefit from the knowledge locked in the data [3][4].

An efficient query routing aims to limit the consumption of network bandwidth by reducing messages across the network and also to reduce the total query processing cost by minimizing the number of peers that contribute to the query's results. Finally, routing in P2P networks is crucial for the scalability of the network.

Wolfgang Nejdl et. al in [5][6][7] presented the routing approach based on routing indices. This approach has been suggested and adapted under various scenarios. It is built upon an RDF-based peer-to-peer network. Queries and answers of queries are represented using RDF metadata which we can use together with the RDF metadata to describe the content of peers in order to build explicit routing indices that facilitate more sophisticated routing approaches.

The advanced technique of [8][9] is also applied for Super-Peer Schema-Based peer-to-peer networks. Based on predefined policies, a fully decentralized broadcast and matching approach distributes the peers automatically to super-peers. The basic idea here is that the super-peer establishes and maintains a specific Semantic Overlay Cluster (SOC). Comparing to our approach, our proposed architecture is build by regrouping the super-peers according to their interest with integrating in each group an index (decision tree) to find the relevant super-peer and other groups in an intelligent way.

Raahemi, Hayajneh and Rabinovitch [10] present a new approach using data-mining technique. In particular they used a decision tree, to classify peer-to-peer (P2P) traffic in IP networks by capturing Internet traffic at a main gateway router, they also performed preprocessing on the data, selected the most significant attributes, and prepared a training-data set to which the decision-tree algorithm was applied. By detecting communities of peers, we achieved classification accuracy of higher than

98 %. However, our approach uses data-mining (decision tree) to classify the super-peers (communities). By detecting communities of peers, we achieved classification accuracy of higher than 99 %.

Bhaduri, Wolff, Giannella and Kargupta [11] propose a P2P decision tree induction algorithm in which every peer learns and maintains the correct decision tree compared to a centralized scenario. This algorithm is completely decentralized, asynchronous, and adapts smoothly to changes in the data and the network. Odysseas Papapetrou [12] proposes new approaches for enabling distributed IR over P2P without limiting the network size or mutilating the IR.

Nottelmann and Fuhr [13] build an IR system over a hierarchical P2P network. The peers there do not maintain a distributed index; instead, some super-peers are assigned the responsibility to keep their peers' summaries, and to forward the queries to the most related of their peers, or to other super-peers.

Sharma and al. [14] introduce a system, called IR-Wire, for information retrieval research in the peer-to-peer file-sharing domain. This tool maintains many statistics, implements a number of information retrieval ranking functions and contains a data logger and analyzer. The data analyzer provides a simple user interface for data analysis. This work was meant to address in the research for tools and data for P2P IR, expressed in [15].

Today's, data management in peer-to-peer (P2P) provide a promising approach that offers scalability, adaptively to high dynamics, and failure resilience. Although there exist many P2P data management systems in the literature, most of them focus on providing only information retrieval (IR) [16][17][18][19] or filtering (IF) [20] functionality (also referred to as publish/subscribe or alerting), and have no support for a combined service. DHTrie [21] is an exact IR and IF system that stresses retrieval effectiveness, while MAPS [22], [23] provides approximate IR and IF by relaxing recall guarantees to achieve better scalability.

IX. Conclusion

Discovering domains on the fly are essential to perform domain directed searching. We show that while our techniques maintain the better quality of results as currently used techniques, our techniques reduce response time in P2P search (35 % at 1500 peers in DK architecture less then Baseline architecture). The advantage of our technique is the robustness in Queries routing. We experiment our technique using a Java implementation. The experiments involve communication in a large, wide-area cluster computer. We have implemented a new simulator by providing several functions many overlay protocols have in common like execution time, overlay message handling and concerning information retrieval like precision and recall. By analyzing the outcome of the experiments, we demonstrate that the system indeed shows the scalability and dependability properties predicted by our previous theoretical and simulation results. Through scalable design, we have easily achieved to simulate our P2P network with 5000 nodes in a reasonable amount of time. The large number of implemented overlay protocols and the availability to collect various statistical data make our simulator a powerful tool for the peer-to-peer research community. Another major direction for future work is in enhancing more the performance (Answering time) by logical restructure for our P2P network by using the minimum traverse between the super-peers (clusters). When the number of the Knowledge-super-peers increases, we jump to the logical restructure method.

References

[1] S. Ioannidis and P. Marbach, On the Design of Hybrid Peer-to-Peer Systems, SIGMETRICS'08, June 2-6, 2008, Annapolis, Maryland, USA.
[2] S. Annapureddy, S. Guha, C. Gkantsidis, D. Gunawardena and P. R. Rodriguez, Is high-quality VoD feasible using P2P swarming? In WWW, 903-912, 2007.
[3] K. Liu, K. Bhaduri, K. Das, P. Nguyen and H. Kargupta, "Client-side Web Mining for Community Formation in Peer-to-Peer Environments," SIGKDD Explorations, vol. 8, no. 2, pp. 11-20, 2006. 2. K. Das, K.
[4] K. Bhaduri and H. Kargupta, "Distributed Identification of Top-l Inner Product Elements and its Application in a Peer-to-Peer Network," IEEE Transactions on Knowledge and Data Engineering (TKDE), vol. 20, no. 4, pp. 475-488, 2008.
[5] W. Nejdl, M. Wolpers, W. Siberski, A. Laser, I. Bruckhorst, M. Schlosser, and C. Schmitz, Super-Peer-Based Routing and Clustering Strategies for RDF-Based Peer-To- Peer Networks. In Proceedings of the 12th International World Wide Web Conference (WWW2003), Budapest, Hungary, May 2003.
[6] W. Nejdl, M. Wolpers, W. Siberski, A. Laser, I. Bruckhorst, M. Schlosser and C. Schmitz, Super-Peer-Based Routing Strategies for RDF-Based P2P Systems. In Proceedings of the 2nd International Workshop On Databases, Information Systems and Peer-to-Peer Computing, Toronto, Canada, September 2004.
[7] W. Nejdl, M. Schlosser, W. Siberski, M. Wolpers, B. Simon, S. Decker and M. intek, RDF-based Peer-To-Peer-Networks for Distributed (Learning) Repositories. Technical Report, November 2002.
[8] A. Laser, M. Wolpers, W. Siberski and W. Nejdl, Efficient data store discovery in a scientific P2P network. In International Workshop on Semantic Web Technologies for Searching and Retrieving Scientific Data, In Proceedings of the ISWC 2003, Florida, USA, October 2003.
[9] A. Laser, F. Naumann, W. Siberski, W. Nejdl and U. Thaden, Semantic overlay clusters within super-peer networks. In Proceedings of the International Workshop on Databases, Information Systems and Peer-to-Peer Computing in Conjunction with the VLDB 2003, Berlin, Germany, September 2003.
[10] B. Raahemi, A. Hayajneh and P. Rabinovitch, Peer-to-Peer IP Traffic Classification Using Decision, International Journal of Business Data Communications and Networking, Volume 3, Issue 4, edited by Jairo Gutierrez © 2007, IGI Global.
[11] K. Bhaduri, R. Wolff, C. Giannella and H. Kargupta, Distributed Decision Tree Induction in Peer-to-Peer Systems, Statistical Analysis and Data Mining Journal (accepted in press), 2008.
[12] O. Papapetrou, Full-text indexing and Information Retrieval in P2P Systems, ACM International Conference Proceeding Series; Vol. 326, EDBT, Nantes, France, Pages 49-57, 2008.

[13] H. Nottelmann and N. Fuhr, A decision-theoretic model for decentralised query routing in hierarchical peer-to-peer networks. In ECIR, pages 148-159, 2007.
[14] S. Sharma, L. T. Nguyen, D. Jia, IR-Wire: A Research Tool for P2P Information Retrieval. In Proc. ACM Wkshp. Open Source Inf. Retr., 2006.
[15] H. Nottelmann, K. Aberer, J. Callan and W. Nejdl, CIKM 2005 P2PIR Workshop Report, 2005, http://p2pir.is.informatik.uni-duisburg.de/2005/report.pdf
[16] M. Bender, S. Michel, P. Triantafillou, G. Weikum and C. Zimmer, MINERVA: Collaborative P2P Search. In VLDB, 2005.
[17] J. Lu and J. Callan, Federated Search of Text-based Digital Libraries in Hierarchical Peer-to-Peer Networks. In ECIR, 2005.
[18] J. Stribling, I. Councill, J. Li, M. Kaashoek, D. Karger, R. Morris and S. Shenker, Overcite: A Cooperative Digital Research Library. In IPTPS, 2005.
[19] M. Bender, S. Michel, P. Triantafillou, G. Weikum and C. Zimmer, Improving Collection Selection with Overlap-Awareness. In SIGIR, 2005.
[20] I. Aekaterinidis and P. Triantafillou, PastryStrings: A Comprehensive Content- Based Publish/Subscribe DHT Network. In ICDCS, 2006.
[21] C. Tryfonopoulos, S. Idreos and M. Koubarakis, Publish/Subscribe Functionality in IR Environments using Structured Overlay Networks. In SIGIR, 2005.
[22] C. Zimmer, C. Tryfonopoulos, K. Berberich, M. Koubarakis and G. Weikum, Approximate Information Filtering in Peer-to-Peer Networks. In WISE, 2008.
[23] C. Zimmer, C. Tryfonopoulos and G. Weikum, MinervaDL: An Architecture for Information Retrieval and Filtering in Distributed Digital Libraries. In ECDL, 2007.
[24] D. Faye, Médiation de données sémantique dans SenPeer, un système pair-à-pair de gestion de données, PHD Thesis, Nantes, France, October 2007.
[25] Haldun Aytug. http://www.cs.waikato.ac.nz/ml/weka, 2006.
[26] F. Howell and R. McNab (1998) "simjava: a discrete event simulation package for Java with applications in computer systems modelling", in proc. First International Conference on Web-based Modelling and Simulation, San Diego CA, Society for Computer Simulation, 1998.
[27] J.P. Erard, livre : Simulation par événements discrets, 1996.
[28] I. Clarke, O.Sandberg, B. Wiley and T. W. Hong, Freenet: A distributed anonymous information storage and retrieval system. Freenet White Paper, 1999. [Online]. Available: http://freenetproject.org/freenet.pdf.
[29] A. Montresor, H. Meling and Ö. Babaoglu, Towards adaptive, resilient and selforganizing peer-to-peer systems. In Revised Papers from the NETWORKING 2002 Workshops on Web Engineering and Peer-to-Peer Computing, London, UK, Springer-Verlag, pp. 300–305, 2002.
[30] C. Tempich, J. Tane, S. Staab, M. Ehrig and C. Schmitz, Towards evaluation of peer-to-peer-based distributed information management systems. In L. van Elst et al. (EDS.), réd., Agent-mediated Knowledge Management - AMKM-2003, AAAI Spring Symposium 2003, Stanford, pp. 73–88, 2003.
[31] J. Li, J. Stribling, R. Morris, M. Kaashoek, and T. Gil, "A performance vs. cost framework for evaluating DHT design tradeoffs under churn," in INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE, vol. 1, Mar. 2005, pp. 225–236.
[34] M. F. Kaashoek and D. R. Karger, "Koorde: A simple degreeoptimal distributed hash table," in Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03), vol. Volume 2735/2003, 2003, pp. 98–107.
[35] P. Maymounkov and D. Mazires, "Kademlia: A peer-to-peer information system based on the xor metric," in Peer-to-Peer Systems: First InternationalWorkshop, IPTPS 2002 Cambridge, MA, USA, March 7-8, 2002. Revised Papers, vol. Volume 2429/2002, 2002, pp. 53–65.
[36] K. Shudo. Overlay weaver. [Online]. Available: http:// overlayweaver.sourceforge.net/

Anis ISMAIL did his PHD on P2P networks in Aix-Marseille II University, France. He works as system and network administrator and instructor at Lebanese University, University institute of technology, saida, Lebanon. His Interest domain concerns Arabic language processing and data management in peer-to- peer systems.

Mohamed QUAFAFOU did his PhD Thesis in 1992 on Intelligent Tutoring Systems at INSA de Lyon, France. From 1992 to 1994, he was ATER at INSA de Lyon and than at Nantes Faculty of Sciences. From 1995 to 2001, he was assistant professor at the Nantes University. During that period, he developed research on Rough Set Theory, concepts approximation, data mining, web information extraction and participated actively with France Telecom to the project Comminges to design a new web system dedicated to French web analysis for discovering emergent web communities. He was also chief-scientist at GEOBS where he headed the Geobs Data Analyzer project, which was developing a spatial data mining systems with application to environment, marketing, social analysis, etc. From September 2002, he was professor at the Avignon University and moved in 2005 to the Aix-Marseille University where he joined the Information and System Science Laboratory (UMR CNRS 6168) and continue his research on web data mining considering different application contexts like P2P, multimedia and web services. Since 2002, he teaches foundations of data/knowledge based systems including machine learning, data mining, personalization, datawarehousing, XML, web services, multimedia, web and mobile applications.

Gilles NACHOUKI is assistant professor at Nantes University in France. He received a PhD in Computer Science from the University of Toulouse. His interest domain concerns distributed databases. Currently his principal works lie in the domain of data management in peer-to- peer systems.

Mohammad HAJJAR is a Professor at University Institute of Technology, Lebanese University, in Lebanon. He received a PHD in computer Science at Nantes University in France. His Interest domain concerns Arabic language processing, multimedia information research and data management in peer-to-peer systems.