Information Searching and Sharing in Large-Scale Dynamic Networks

George A. Vouros

Department of Information & Communication Systems Engineering
University of the Aegean, Samos, Greece
georgev@aegean.gr

ABSTRACT

Finding the right agents in a large and dynamic network to provide the needed resources in a timely fashion, is a long standing problem. This paper presents a method for information searching and sharing that combines routing indices with token-based methods. The proposed method enables agents to search effectively by acquiring their neighbors' interests, advertising their information provision abilities and maintaining indices for routing queries, in an integrated way. Specifically, the paper demonstrates through performance experiments how static and dynamic networks of agents can be 'tuned' to answer queries effectively as they gather evidence for the interests and information provision abilities of others, without altering the topology or imposing an overlay structure to the network of acquaintances.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence – *multiagent systems*

Keywords

Artificial social systems, Performance, Scalability, Robustness, and Dependability

1. INTRODUCTION

Considering to be a decentralized control problem, information searching and sharing in large-scale systems of cooperative agents is a hard problem in the general case: The computation of an optimal policy, when each agent possesses an approximate partial view of the state of the environment and when agents' observations and activities are interdependent (i.e. one agent's actions affect the observations and the state of an other) [3], is hard. This fact, has resulted to efforts that either require agents to have a global view of the system [15], to heuristics [4], to pre-computation of agents' information needs and information provision capabilities for proactive communication [17], to localized reasoning processes built on incoming information [12,13,14], and to mathematical frameworks for coordination whose optimal policies can be approximated [11] for small (sub-) networks of associated agents.

On the other hand, there is a lot of research on semantic peer to peer search networks and social networks [1,5,6,8,9,10,16,18,19] many of which deal with tuning a network of peers for effective information searching and sharing. They do it mostly by imposing logical and semantic overlay structures. However, as far as we know there is no work that demonstrates the effectiveness of a gradual tuning process in large-scale dynamic networks that studies the impact of the information gathered by agents as more and more queries are issued and served in concurrent sessions in the network.

The main issue in this paper concerns 'tuning' a network of agents, each with a specific expertise, for efficient and effective information searching and sharing, without altering the topology or imposing an overlay structure via clustering, introduction of shortcut indices, or re-wiring. '*Tuning*' is the task of sharing and gathering the *necessary* knowledge for agents to propagate requests to the *right* acquaintances, minimizing the searching effort, increasing the efficiency and the benefit of the system. Specifically, this paper proposes a method for information searching and sharing in dynamic and large scale networks, which combines routing indices with token-based methods for information sharing in large-scale multi-agent systems.

This paper is structured as follows: Section 2 presents related work and motivates the proposed method. Section 3 states the problem and section 4 presents in detail the individual techniques and the overall proposed method. Section 5 presents the experimental setup and results, and section 6 concludes the paper, sketching future work.

2. RELATED WORK

Information provision and sharing can be considered to be a decentralized partially-observable Markov decision process [3,4,11,14]. In the general case, decentralized control of large-scale dynamic systems of cooperative agents is a hard problem. Optimal solutions can only be approximated by means of heuristics, by relaxations of the original problem or by centralized solutions. The computation of an optimal control policy is simple given that global states can be factored, that the probability of transitions and observations are independent, the observations combined determine the global state of the system and the reward function can be easily defined as the sum of local reward functions [3].

However, in a large-scale dynamic system with decentralized control it is very hard for agents to possess accurate partial views of the environment, and it is even more hard for agents to possess

AAMAS'07, May 14-18, 2007, Honolulu, Hawai'i, USA.

a global view of the environment. Furthermore, agents' observations can not be assumed independent, as one agent's actions can affect the observations of others: For instance, when one agent joins/leaves the system, then this may affect other agents' assessment of neighbours' information provision abilities. Furthermore, the probabilities of transitions can be dependent too; something that increases the complexity of the problem: For example, when an agent sends a query to another agent, then this may affect the state of the latter, as far as the assessed interests of the former are concerned.

Considering independent activities and observations, authors in [4] propose a decision-theoretic solution treating standard action and information exchange as explicit choices that the decision maker must make. They approximate the solution using a myopic algorithm. Their work differs in the one reported here in the following aspects: First, it aims at optimizing communication, while the goal here is to tune the network for effective information sharing, reducing communication and increasing system's benefit. Second, the solution is approximated using a myopic algorithm, but authors do not demonstrate how sub-optimal are the solutions computed (something we neither do), given their interest to the optimal solution. Third, they consider that transitions and observations made by agents are independent, which, as already discussed, is not true in the general case. Last, in contrast to their approach where agents broadcast messages, here agents decide not only when to communicate, but to whom to send a message too.

Token based approaches are promising for scaling coordination and therefore information provision and sharing to large-scale systems effectively. In [11] authors provide a mathematical framework for routing tokens, providing also an approximation to solving the original problem in case of independent agents' activities. The proposed method requires a high volume of computations that authors aim to reduce by restricting its application to static logical teams of associated agents. In accordance to this approach, in [12,13,14], information sharing is considered only for static networks and self-tuning of networks is not demonstrated. As it will be shown in section 5, our experiments show that although these approaches can handle information sharing in dynamic networks, they require a larger amount of messages in comparison to the approach proposed here and can not tune the network for efficient information sharing.

Proactive communication has been proposed in [17] as a result of a dynamic decision theoretic determination of communication strategies. This approach is based on the specification of agents as "providers" and "needers": This is done by a plan-based pre-computation of information needs and provision abilities of agents. However, this approach can not scale to large and dynamic networks, as it would be highly inefficient for each agent to compute and determine its potential needs and information provision abilities given its potential interaction with 100s of other agents.

Viewing information retrieval in peer-to-peer systems from a multi-agent system perspective, the approach proposed in [18] is based on a language model of agents' documents collection. Exploiting the models of other agents in the network, agents construct their view of the network which is being used for forming routing decisions. Initially, agents build their views using the models of their neighbours. Then, the system reorganizes by forming clusters of agents with similar content. Clusters are being exploited during information retrieval using a kNN approach and a gradient search scheme. Although this work aims at tuning a network for efficient information provision (through re-organization), it does not demonstrate the effectiveness of the approach with respect to this issue. Moreover, although during re-organization and retrieval they measure the similarity of content between agents, a more fine grained approach is needed that would allow agents to measure similarities of information items or sub-collections of information items. However, it is expected that this will complicate re-organization. Based on their work on peer-to-peer systems, H.Zhand and V.Lesser in [19] study concurrent search sessions. Dealing with static networks, they focus on minimizing processing and communication bottlenecks: Although we deal with concurrent search sessions, their work is orthogonal to ours, which may be further extended towards incorporating such features in the future.

Considering research in semantic peer-to-peer systems[1], most of the approaches exploit what can be loosely stated a "routing index". A major question concerning information searching is "what information has to be shared between peers, when, and what adjustments have to be made so as queries to be routed to trustworthy information sources in the most effective and efficient way".

REMINDIN' [10] peers gather information concerning the queries that have been answered successfully by other peers, so as to subsequently select peers to forward requests to: This is a lazy learning approach that does not involve advertisement of peer information provision abilities. This results in a tuning process where the overall recall increases over time, while the number of messages per query remains about the same. Here, agents actively advertise their information provision abilities based on the assessed interests of their peers: This results in a much lower number of messages per query than those reported in REMINDIN'.

In [5,6] peers, using a common ontology, advertise their expertise, which is being exploited for the formation of a semantic overlay network: Queries are propagated in this network depending on their similarity with peers' expertise. It is on the receiver's side to decide whether it shall accept or not an advertisement, based on the similarity between expertise descriptions. According to our approach, agents advertise selectively their information provision abilities about specific topics to their neighbours with similar information interests (and only to these). However, this is done as time passes and while agents' receive requests from their peers.

The gradual creation of overlay networks via re-wiring, shortcuts creation [1,8,16] or clustering of peers [17,9] are tuning approaches that differ fundamentally from the one proposed here: Through local interactions, we aim at tuning the network for efficient information provision by gathering routing information gradually, as queries are being propagated in the network and agents advertise their information provision abilities given the

[1] General research in peer-to-peer systems concentrates either on network topologies or on distribution of documents: Approaches do not aim to optimize advertising, and search mostly requires common keys for nodes and their contents. They generate a substantial overhead in highly dynamic settings, where nodes join/leave the system.

interests of their neighbours. Given the success of this method, we shall study how the addition of logical paths and gradual evolution of the network topology can further increase the effectiveness of the proposed method.

3. PROBLEM STATEMENT

Let $N=\{A_1, A_2, ...A_n\}$ be the set of agents in the system. The network of agents is modelled as a graph $G=(N,E)$, where N is the set of agents and E is a set of bidirectional edges denoted as non-ordered pairs (A_i,A_j). The neighbourhood of an agent A_i includes all the one-hop away agents (i.e. its acquaintance agents) A_j such that $(A_i, A_j)\in E$. The set of acquaintances of A_i is denoted by $N(A_i)$.

Each agent maintains (a) an ontology that represents categories of information, (b) indices of information pieces available to its local database and to other agents, and (c) a profile model for some of its acquaintances. Indices and profile models are described in detail in section 4.

Ontology concepts represent categories that classify the information pieces available. It is assumed that agents in the network share the same ontology, but each agent has a set of information items in its local repository, which are classified under the concepts of its expertise. The set of concepts is denoted by C. It is assumed that the sets of items in agents' local repositories are non-overlapping.

Finally, it is assumed that there is a set of k queries $T=\{t_1,..., t_k\}$. Each query is represented by a tuple *<id,a,c,path,ttl>,* where *id* is the unique identity of the query*, a* is a non-negative integer representing the maximum number of information pieces requested, *c* is the specific category to which these pieces must belong, *path* is a path in the network of agents through which the query has been propagated (initially it contains the originator of the query and each agent appends its id in the *path* before propagating the query), and *ttl* is a positive integer that specifies the maximum number of hops that the query can reach. In case this limit is exceeded and the corresponding number of information pieces have not been found, then the query is considered "unfulfilled" However, even in this case, a (possibly high) percentage of the requested pieces of information may have been found.

The problem that this article deals with is as follows: Given a network of agents $G=(N,E)$ and a set of queries T, agents must retrieve the pieces of information requested by queries, in concurrent search sessions, and further 'tune' the network so as to answer future similar queries in the more effective and efficient way, increasing the benefit of the system and reducing the communication messages required. The *benefit* of the system is the ratio of information pieces retrieved to the number of information pieces requested. The *efficiency* of the system is measured by the number of messages needed for searching and updating the indexes and profiles maintained.

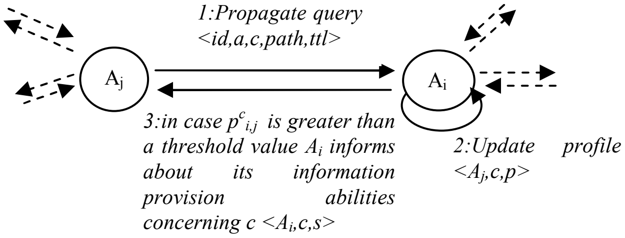
'Tuning' the network requires agents to acquire the necessary information about acquaintances' interests and information provision abilities (i.e. the routing and profiling tuples detailed in section 4), so as to route queries and further share information in the most efficient way. This must be done seamlessly to searching: I.e. agents in the network must share/acquire the necessary information while searching, increasing the benefit and efficiency gradually, as more queries are posed.

4. INFORMATION SEARCHING AND SHARING

4.1 Overall Method

Given a network $G=(N,E)$ of agents and a set of queries T, each agent maintains indices for routing queries to the "right agents", as well as acquaintances' profiles for advertising its information provision abilities to those interested.

To capture information about pieces of information accessible by the agents, each agent A maintains a routing index that is realized as a set of tuples of the form $<A_i,c,s>$. Each such tuple specifies the number s of information items in category c that can be reached by *A through* A_i, such that $A_i\in N(A)\cup\{A\}$. This specifies the *information provision abilities* of A_i to A with respect to the information category c. As it can be noticed, each tuple corresponds either to the agent A itself (specifying the pieces of information classified in c available to its local repository) or to an acquaintance of the agent (recording the pieces of information in category c available to the acquaintance agent and to agents that can be reached through this acquaintance). The routing index is exploited for the propagation of queries to the "right" agents": Those that are either more likely to provide answers or that know someone that can provide the requested pieces of information.


1:Propagate query
<id,a,c,path,ttl>
A_j
A_i
3:in case $p^c_{i,j}$ is greater than a threshold value A_i informs about its information provision abilities concerning c $<A_i,c,s>$
2:Update profile $<A_j,c,p>$

Considering an agent A_i, the profile model of some of its acquaintances A_j, denoted by P_{ij}, is a set of tuples $<A_j,c,p>$, maintained by A_i. Such a tuple specifies the probability p that the acquaintance A_j is interested to pieces of information in category c (subsequently, such a probability is also denoted by $p^c_{i,j}$). Formally, the profile model of an acquaintance A_j of A_i is $P_{ij}=\{<A_j, c, p^c_{i,j}>|A_j\in N(A_i)$ and $c\in C\}$. Profile models are exploited by the agents to decide where to 'advertise' their information provision abilities.

Figure 1. Typical pattern for information sharing between two acquaintances (numbers show the sequence of tasks)

Given two acquaintances A_i and A_j in G, the information searching and sharing process proceeds as it is depicted in Figure 1: Initially, each agent has no knowledge about the information provision abilities of its acquaintances and also, it possesses no information about their interests. When that the query *<id,a,c,path,ttl>* is sent to A_i from the agent A_j, then A_i has to update the profile of A_j concerning the category *c,* increasing the probability $p^c_{i,j}$ that A_j is interested to information in c. When this probability is greater than a threshold value (due to the queries about c that A_j has sent to A_i), then A_i assesses that it is highly probable for A_j to be interested about information in category c.

This leads A_i to inform A_j about its information provision abilities as far as the category c is concerned. This information is being used by A_j to update its index about A_i. This index is being exploited by A_j to further propagate queries, and it is further propagated to those interested in c. Moreover, the profile of A_j maintained by A_i guides A_i to propagate changes concerning its information provision abilities to $A_{j.}$

The above method has the following features: (a) It combines routing indices and token-based information sharing techniques for efficient information searching and sharing, without imposing an overlay network structure. (b) It can be used by agents to adapt safely and effectively to dynamic networks. (c) It supports the acquisition and exploitation of different types of locally available information for the 'tuning' process. (d) It extends the token-based method for information sharing (as it was originally proposed in [12,13]) in two respects: First, to deal with categories of information represented by means of ontology concepts and not with specific pieces of information, and second, to guide agents to advertise information that is semantically similar to the information requested, by using a semantic similarity measure between information categories. Therefore, it paves the way for the use of token-based methods for semantic peer-to-peer systems. This is further described in section 4.3. (d) It provides a more sophisticated way for agents to update routing indices than that originally proposed in [2]. This is done by gathering and exploiting acquaintances' profiles for effective information sharing, avoiding unnecessary and cyclic updates that may result to misleading information about agents' information provision abilities. This is further described in the next sub-section.

4.2 Routing Indices

As already specified, given a network of agents $G=(N,E)$, and the set $N(A)$ of agent's A acquaintances, the routing index (RI) of A (denoted by $RI(A)$) is a collection of at most $|C|\cdot|N(A)\cup\{A\}|$ indexing tuples $<A_k,c,s>$. The key idea is that given such an index and a request concerning c, A will forward this request to A_k if the resources available *via* A_k (i.e. the information abilities of A_k to A) can best serve this request. To compute the information abilities of A_k to A, all tuples $<A_j,c,s_i>$ concerning all agents in $N(A_k)$-$\{A\}$ must be aggregated. Crespo and Garcia-Molina [2] examine various types of aggregations. In this paper, given some tuples $<A_{i1},c,s_{11}>,<A_{21},c,s_{21,}\ldots>$ maintained by the agent A_k, their aggregation is the tuple $<A_k,c,sum(s_{11},s_{21}\ldots)>$. This gives information concerning the pieces of information that can be provided through A_k, but it does not distinguish what each of A_k's acquaintances can provide: This is an inherent feature of routing indices. Without considering the interests of its acquaintances, A_k may compute aggregations concerning agents in $N(A)\cup\{A\}$-$\{A_i\}$ and advertise/share its information provision abilities to each agent A_i in $N(A)$.

For instance, given the network configuration depicted in Figure 2 and a category c, agent A_k sends the aggregation of the tuples concerning agents in $N(A_k)\cup\{A\}$-$\{A_2\}$ (denoted as *aggregation*(A_k,A_1,c)) to agent $A_{2,}$ which records the tuple $<A_k,c,X>$. Similarly the aggregation of the tuples concerning the agents in $N(A_k)\cup\{A_k\}$-$\{A_1\}$ (denoted as *aggregation*(A_k,A_2,c)) is sent to the agent A_1, which also records the tuple $<A_k,c,s>$. It must be noticed that A_1 and A_2 record the information provision abilities of A_k each from "its own point of view". Every time the tuple that models the information provision abilities of an agent changes, the aggregation has to re-compute and send the new aggregation to the appropriate neighbors in the way described above. Then, its neighbors have to propagate these updates to their acquaintances, and so on.

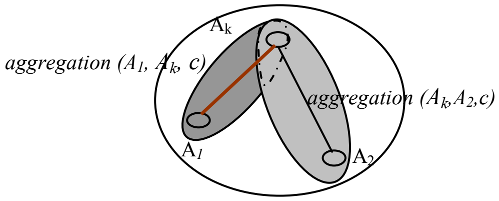


Figure 2.**Aggregating and sharing information provision indices**.

Routing indices may be misleading and lead to inefficiency in arbitrary graphs containing cycles. The exploitation of acquaintances' profiles can provide solutions to these deficiencies. Each agent propagates its information provision abilities concerning a category c only to these acquaintances that have high interest in this category. As it has been mentioned, an agent "expresses" its interest in a category by propagating queries about it. Therefore, indices concerning a category c are propagated in the inverse direction in the paths to which queries about c are propagated. Indices are propagated as long as agents in the path have a high interest in c. Queries can not be propagated in a cyclic fashion since an agent serves and propagates queries that have not been served by it in a previous time point. Therefore, due to their relation to queries, indices are not propagated in a cyclic fashion, as well. However, there is still a specific case where cycles can not be avoided. Such a case is shown in Figure 3:

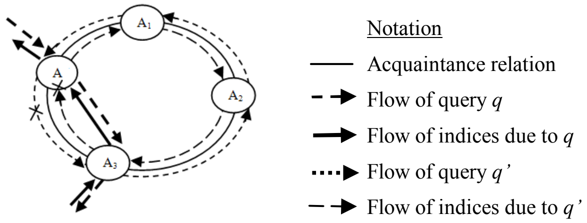


Figure 3. Cyclic pattern for the sharing of indices.

While the propagation of the query *q'* causes the propagation of information provision abilities of agents in a non cyclic way (since the agent A recognizes that *q'* has been served), the query *q* causes the propagation of information abilities of A to other agents in the network, causing, in conjunction to the propagation of indices due to *q'*, a cyclic update of indices.

4.3 Profiles

The key assumption behind the exploitation of acquaintances' profiles, as it was originally proposed in [12,13], is that for an agent to pass a specific information item, this agent has a high interest on it or to related information. As already said, in our case, acquaintances' profiles are created based on received queries and specify the interests of acquaintances to specific information categories. Given the query $<id,a,c,path,ttl>$ sent from A_j to A_i, A_i has to record not only the interest of A_j to c, but the interest of A_j to all the related classes, given their semantic similarity to c.

To measure the similarity between two ontology classes we use the similarity function $sim{:}C{\times}C{\rightarrow}[0,1]$ [7]:

$$sim(c_i,c_j) = \begin{cases} e^{-al} \bullet \dfrac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}} & \text{if } c_i \text{ is a subconcept of } c_j \\ 1 & c_i = c_j \\ 0.01 & \text{otherwise} \end{cases}$$

where l is the length of the shortest path between c_i and c_j in the graph spanned by the sub concept relation and h the minimal level in the hierarchy of either c_i,or c_j. α and β are parameters scaling the contribution of shortest path length l and h, respectively. Based on previous works we choose α=0.2 and β=0.6 as optimal values. It must be noticed that we measure similarity between sub-concepts, assigning a very low similarity value between concepts that are not related by the sub-concept relation. This is due to that, each query about information in category c_i can be answered by information in any sub-category of c_i close enough to c_i. Given a threshold value 0.3, $sim(c_i,c_j)\geq 0.3$ indicates that an agent interested in c_i is also interested in c_j, while $sim(c_i,c_j)<0.3$ indicates that an agent interested in c_i is unlikely to be interested in c_j. This threshold value was chosen after some empirical experiments with ontologies.

The update of A_i's assessment on $p^c_{i,j}$ based on an incoming query $<id,a,c,path,ttl>$ from A_j is computed by leveraging Bayes Rule as follows [12,13]:

$\forall c_k \in C$, and $A \in N(A_i)$

$$\delta_p = \begin{cases} p_{ij}^{c_k} \times sim(c,c_k) \times \dfrac{2}{|N|} & \text{If } A_j \text{ is the last in the } path \\ p_{ij}^{c_k} \times \dfrac{1}{|N|} & \text{If } A_j \text{ is not the last in the } path \end{cases}$$

Then probabilities must be normalized to ensure that

$$\forall c \in C, \quad \sum_{A_j \in N(A_i)} p_{i,j}^{c} = 1$$

According to the first case of the equation, the probability that the agent that has propagated a query about c to be interested about information in c_k, is updated based on the similarity between c and c_k. The second case updates the interests of agents other than the requesting one, in a way that ensures that normalization works. It must be noticed that in contrast to [12,13], the computation has been changed in favour to the agent that passed the query.

The profiles of acquaintances enable an agent to decide where and which advertisements to be sent. Specifically, for each $A_j \in N(A_i)$ and $c \in C$ for which $p^c_{i,j}$ is greater than a threshold value (currently set to 0.5), the agent A_i aggregates the vectors (A_k,c,s) of each agent $A_k \in N(A_i) \cup \{A_i\} - \{A_j\}$and sends the tuple (A_i,c,s') to A_j. Also, given a high $p^c_{i,j}$, when a change to an index concerning c occurs (e.g. due to a change in A_i's local repository, or due to that the set of its acquaintances changed), A_i sends the updated aggregated index entry to A_j. Doing so, the agent A_j which is highly interested to pieces of information in category c updates its index so as to become aware of the information provision abilities of A_i as far as the category c is concerned.

4.4 Tuning

Tuning is performed seamlessly to searching: As agents propagate queries to be served, their profiles are getting updated by their acquaintances. As their profiles are getting updated, agents receive the aggregated indices of their acquaintances, becoming aware of their information provision abilities on information categories to which they are probably interested. Given these indices, agents further propagate queries to acquaintances that are more likely to serve queries, and so on. Concerning the routing index and the profiles maintained by an agent A, it must be pointed that A does not need to record all possible tuples, i.e. $|C| \cdot |N(A) \cup \{A\}|$: It records only those that are of particular interest for searching and sharing information, depending on the expertise and interests of its own and its acquaintances.

Initially, agents do not possess profiles of their acquaintances. For indices there are two alternatives: Either agents do not initially possess any information about acquaintances' local repositories (this is the "*no initialization of indices*" case), or they do (this is the "*initialization of indices*" case). Given a query, agents propagate this query to those acquaintances that have the highest information provision abilities. In the "no initialization of indices" case where an agent does not initially possess information about its acquaintances' abilities, it may initially propagate a query to all of them, resulting to a pure flooding approach; or it may propagate the query randomly to a percentage of them. In the "initialization of indices" case, where an agent initially possesses information about its acquaintances' local repository, it can propagate queries to all or to a percentage of those that can best serve the request. We considered both cases in our experiments.

Given a static setting where agents do not shift their expertise, and the distribution of information pieces does not change, the network will eventually reach a state where no information concerning agents' information abilities will need to be propagated and no agents' profiles will need to be updated: Queries shall be propagated only to those agents that will lead to a near-to-the-maximum benefit of the system in a very efficient way. In a dynamic setting, agents may shift their expertise, their interests, they may leave the network at will, or welcome new agents that join the network and bring new information provision abilities, new interests and new types of queries. In this paper we study settings where agents may leave or join the network. This requires agents to adapt safely and effectively. Towards this goal, in case an agent does not receive a reply from one of its acquaintances within a time interval, then it retracts all the indices and the profile concerning the missing acquaintance and re-propagates the queries that have been sent to the missing agent since the last successful handshake, to other agents. In case a new agent joins the network, then its acquaintances that are getting aware of its presence propagate all the queries that have processed by them in the last n time points (currently n is set to 6) to the newcomer. This is done so as to inform the newcomer about their interests and initiate information sharing.

5. EXPERIMENTAL SETUP

To validate the proposed approach we have built a prototype that simulates large networks. To test the scalability of our approach we have run several experiments with various types of networks. Here we present results from 3 network types with $|N|$=100, $|N|$=500 and $|N|$=1000 that provide representative cases. Networks are constructed by distributing randomly |*N*| agents in an *n*×*n* area, each with a "visibility" ratio equal to *r*. The acquaintances of an agent are those that are "visible" to the agent and those from which the agent is visible (since edges in the network are bi-directional). Details about networks are given in Table 1. The column avg(|N(A)|) shows the average number of acquaintances per agent in the network and the column |T| shows the number of queries per network type. It must be noticed that the TypeA network is more "dense" than the others, which are much larger than this.

Each experiment ran 40 times. In each run the network is provided with a new set of randomly generated queries that are originated from randomly chosen agents. The agents search and gather knowledge that they further use and enrich, tuning the network gradually, run by run. Each run lasts a number of rounds that depends on the *ttl* of queries and on the parameters that determine the dynamics of the network: To end a run, all queries must have either been "served" (i.e. 100% of the information items requested must have been found), or they must have been "unfulfilled" (i.e. have exceeded their *ttl*). It must be noticed that in case of a dynamic setting, this ending criterion causes some of the queries to be "lost". This is the case when some queries are the only "active" remained and the agents to whom they have been propagated left the network without their acquaintances to be aware of it.

Table1: **Network types**

	|N|	R	N	avg(|N(A)|)	|T|
TypeA	100	10	25	50	363
TypeB	500	10	125	20	1690
TypeC	1000	10	250	10	3330

Information used in the experiments is synthetic and is being classified in 15 distinct categories: Each agent's expertise comprises a unique information category. For the category in its expertise each agent holds at most 1000 information pieces, the exact number of which is determined randomly.

At each run a constant number of queries are being generated, depending on the type of network used (last column in Table 1). At each run, each query is randomly assigned to an originator agent and is set to request a random number of information items, classified in a sub-category of the query-originator agent's expertise. This sub-category is chosen in a random way and the requested items are less than 6000. The *ttl* for any query is set to be equal to 6. In such a setting, the demand for information items is much higher than the agents' information provision abilities, given the *ttl* of queries: The maximum benefit in any experimental case is much less than 60% (this has been done so as to challenge the 'tuning' task in settings where queries can not be served in the first hop or after 2-3 hops).

Given that agents are initially not aware of acquaintances' local repository ("*no initialization of indices*" case), we have run several evaluation experiments for each network type depending on the percentage of acquaintances to which a query can be propagated by an agent. These types of experiments are denoted by TypeX-Y, where X denotes the type of network and Y the percentage of acquaintances: Here we present results for Y equal to 10, 20 or 50. For instance, TypeA-10 denotes a setting with a network of TypeA where each query is being propagated to at most 10% of an agent's acquaintances. The exact number of acquaintances is randomly chosen per agent and queries are being propagated only to those acquaintances that are likely to best serve the request. Figures 4 and 5 show experiments for static and dynamic networks of TypeA-10 (dense network with a low percentage of acquaintances), TypeB-20 (quite dense network

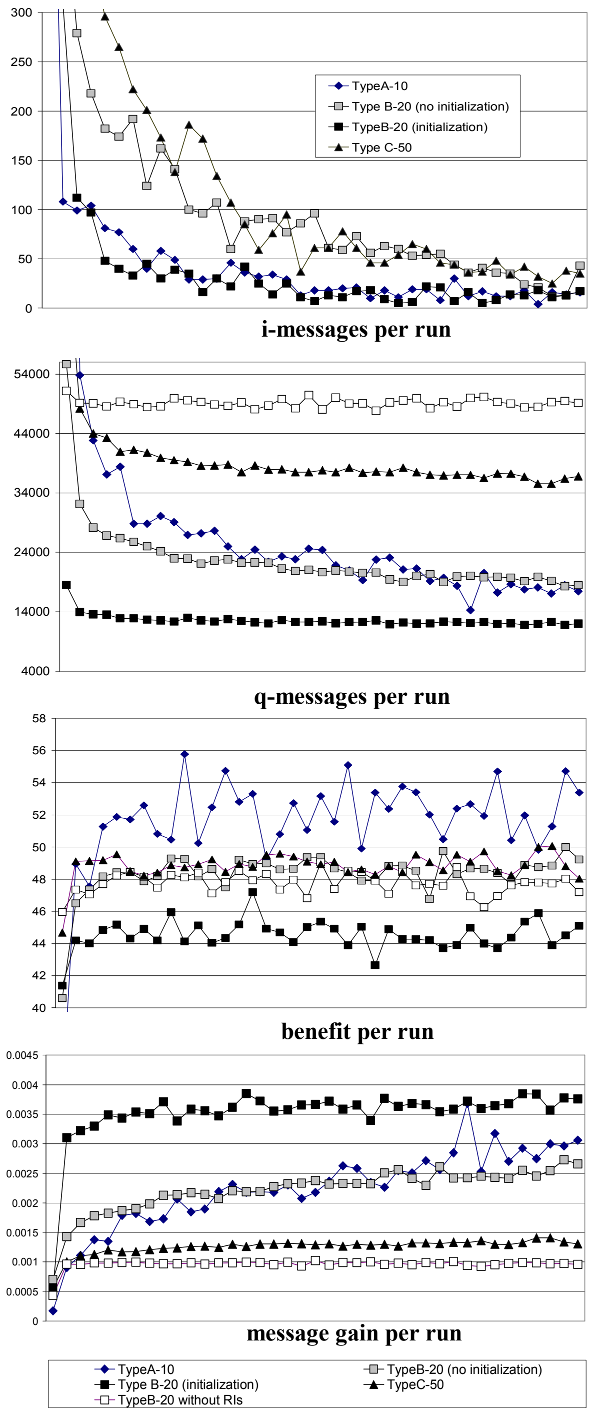


Figure 4. Results for static networks as agents gather information about acquaintances' abilities and interests

with a low percentage of acquaintances), with initialization and without initialization, and TypeC-50 (not a so dense network with a quite high percentage of acquaintances). To demonstrate the advantages of our method we have considered networks without routing indices for TypeC-50 and TypeB-20 networks: Agents in these networks, similarly to [12,13], share information concerning their local repository based on their assessments on acquaintances' interests.

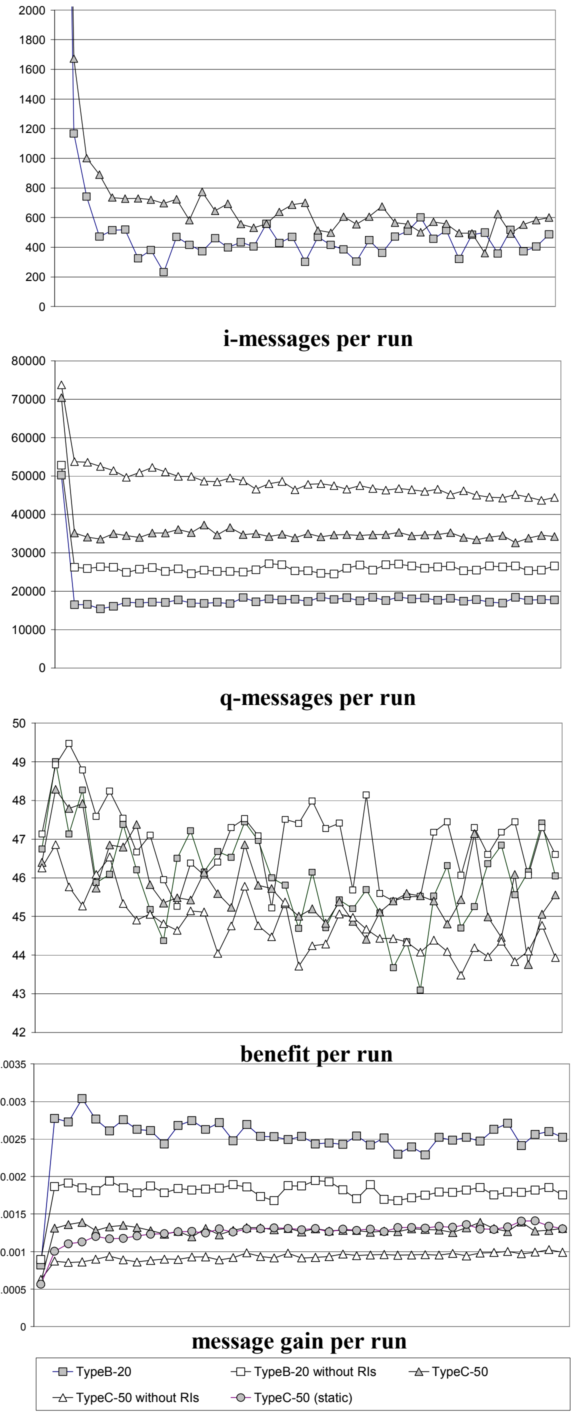


Figure 5. Results for dynamic networks as agents gather information about acquaintances' abilities and interests

Results computed in each experiment show the number of query-propagation messages (*q-messages*), the number of messages for the update of indices (*i-messages*), the *benefit* of the system, i.e. the average ratio of information pieces provided to the number of pieces requested per query, and the *message gain,* i.e. the ratio of benefit to the total number of messages. The horizontal axis in each diagram corresponds to the runs.

As it is shown in Figure 4, as agents search and share information from run 1 to run 40, they manage to increase the benefit of the system, by drastically reducing the number of messages. Also (not shown here due to space reasons) the number of unfulfilled queries decrease, while the served queries increase gradually. Experiments show: (a) An effective tuning of the networks as time passes and more queries are posed to the network, even if agents maintain the models of a small percentage of their acquaintances. (b) That 'tuning' can greatly facilitate the scalability of the information searching and sharing tasks in networks.

To show whether initial knowledge about acquaintances local repository (the "*initialization of indices*" case) affects the effective tuning of the network, we provide representative results from the TypeB-20 network. As it is shown in Figure 4, the tuning task in this case does not manage to achieve the benefit of the system reported for the "*no initialization of indices*" case. On the contrary, while the tuning affects the *i-messages* drastically; the *q-messages* are not affected in the same way: The *q-messages* in the "*initialization of indices"* case are less than those in the TypeB-20 with "*no initialization of indices*" case. This is further shown in a more clear way in the message gain of both approaches: The message gain of the TypeB-20 with "*initialization of indices*" case is higher than the message gain for the TypeB-20 experiment with "*no initialization of indices*". Therefore, initial knowledge concerning local information of acquaintances can be used for guiding searching and tuning at the initial stages of the tuning task, only if we need to gain efficiency (i.e. decrease the number of required messages) to the cost of loosing effectiveness (i.e. have lower benefit): This is due to the fact that, as agents posses information about acquaintances' local repositories, the "tuning" process enables the further exchange of messages concerning agents' information provision abilities *only* in cases where agents' profiles provide evidence for such a need. However, initial information about acquaintances' local repositories may mislead the searching process, resulting in low benefit. In case we need to gain effectiveness to the cost of reducing efficiency, this type of local knowledge does not suffice. Considering also the information sharing method without routing indices ("*without RIs"* cases), we can see that for static networks it requires more *q-messages* without managing to "tune" the system, while the benefit is nearly the same to the one reported by our method. This is shown clearly in the "message gain" diagrams in Figure 4.

Figure 5 provides results for dynamic networks. These are results from a particular representative case of our experiments where more than 25% of (randomly chosen) nodes leave the network in each run during the experiment. After a random number of rounds, a new node may replace the one left. This newcomer has no information about the network. Approximately 25% of the nodes that leave the network are not replaced for 50% of the experiment, and approximately 50% are not replaced for more than 35% of the experiment. In such a highly dynamic setting with very scarce information resources distributed in the network, as Figure 5 shows, the tuning approach has managed to keep the benefit to acceptable levels, while still reducing drastically the number of i-messages. However, as it can be expected, this reduction is not so drastic as it was in the corresponding static cases. Figure 5 shows that the message gain for the dynamic case is comparable to the message gain for the corresponding (TypeC-50) static case, which proves the value of this approach for dynamic settings. The comparison to the case where no routing indices are exploited reveals the same results as in the static case, to the cost of a large number of messages.

Finally it must be pointed that the maximum number of messages per query required by the proposed method is nearly 12, which is less than that that reported by other efforts.

6. CONCLUSIONS

This paper presents a method for semantic query processing in large networks of agents that combines routing indices with information sharing methods. The presented method enables agents to keep records of acquaintances' interests, to advertise their information provision abilities to those that have a high interest on them, and to maintain indices for routing queries to those agents that have the requested information provision abilities. Specifically, the paper demonstrates through extensive performance experiments: (a) How networks of agents can be 'tuned' so as to provide requested information effectively, increasing the benefit and the efficiency of the system. (b) How different types of local knowledge (number, local information repositories, percentage, interests and information provision abilities of acquaintances) can guide agents to effectively answer queries, balancing between efficiency and efficacy. (c) That the proposed "tuning" task manages to increase the efficiency of information searching and sharing in highly dynamic and large networks. (d) That the information gathered and maintained by agents supports efficient and effective information searching and sharing: Initial information about acquaintances information provision abilities is not necessary and a small percentage of acquaintances suffices.

Further work concerns experimenting with real data and ontologies, differences in ontologies between agents, shifts in expertise and the parallel construction of overlay structure.

7. REFERENCES

[1] Cooper, B.F., Garcia-Molina, H. Ad hoc, self-supervising peer-to-peer search networks. *ACM Transactions on Information Systems (TOIS) archive*, Volume 23 ,Issue 2 (April 2005) ,169 - 200

[2] Crespo, A., Garcia-Molina, H. Routing indices for peer-to-peer systems, in *Proc. of the 28th Conference on Distributed Computing Systems*, July 2002.",

[3] Goldman, C., and Zilberstein, S. Decentralized Control of Cooperative Systems: Categorization and Complexity Analysis. *Journal of Artificial Intelligence Research* 22 (2004), 143-174.

[4] Goldman, C., and Zilberstein, S. Optimizing Information Exchange in Cooperative Multi-agent Systems. In *Proc. Proc. of AAMAS 2003*, July 2003.

[5] Haase, P., Siebes, R., van Harmelen, F. Peer Selection in Peer-to-Peer Networks with Semantic Topologies. In *Semantics of a Networked World,* Lecture Notes in Computer Science, Springer, Volume 3226/2004, 108-125.

[6] Haase, P., Broekstra, J., Ehrig, M., Menken, M., Mika, P., Plechawski, M., Pyszlak, P., Schnizler, B., Siebes, R., Staab, S., Tempich, T. Bibster-A Semantics-Based Bibliographic Peer-to-Peer System, In *Proc. of ISWC 2004*, 122–136.

[7] Li, Y., Bandar, Z., and McLean, D.. An approach for measuring semantic similarity between words using semantic multiple information sources. *IEEE Trans. on Knowledge and Data Engineering,* vol. 15, No 4, 2003, 871-882.

[8] Loser, A., Staab, S., Tempich, C. Semantic Social Overlay Networks. *IEEE Journal on Selected Areas in Communications*. To appear 2006/2007

[9] Nejdl, W., Wolpers M., Siberski, W., Schmitz, C., Schlosser, M., Brunkhorst, I., Loser, Super-Peer-Based Routing and Clustering Strategies for RDF-Based Peer-to-Peer Networks.*In Pro. of WWW2003,* 536-543.

[10] Tempich, C., Staab, S., Wranik, A. REMINDIN': Semantic Query Routing in Peer-to-Peer Networks Based on Social Metaphors. In *Proc. of the WWW04,* 640-649.

[11] Xu, Y., Scerri, P., Yu, B., Lewis, M., and Sycara, K. A POMDP Approach to Token-Based Team Coordination. In *Proc. of AAMAS'05*, (July 25-29, Utrecht) ACM Press.

[12] Xu, Y., Lewis, M., Sycara, K., and Scerri, P. Information Sharing in Large Scale Teams. In *AAMAS'04 Workshop on Challenges in Coordination of Large Scale MultiAgent Systems*, 2004.

[13] Xu, Y., Liao, E., Scerri, P., Yu, B., Lewis, M., and Sycara, K. Towards Flexible Coordination of Large Scale Multi-Agent Systems. *In Challenges of Large Scale Coordination* Springer, 2005.

[14] Xu, Y., Scerri, P., Yu, B., Okamoto, S., Lewis, M., and Sycara, K. An Integrated Token Based Algorithm for Scalable Coordination. In *Proc. of AAMAS'05,* 407-414

[15] Xuan, P., Lesser, V., Zilberstein, S. Communication Decisions in Multi-agent Cooperation: Model and Experiments. In *Proc of AGENTS'01,* 2001, 616-623.

[16] Yu, B., and Singh M. Searching Social Networks, In *Proc. of AAMAS'03.*

[17] Zhang, Y., Volz, R., Ioeger, T.R., Yen, J. A Decision Theoretic Approach for Designing Proactive Communication in Multi-Agent Teamwork. In *Proc of SAC'04,*2004, 64-71.

[18] Zhang, H., Croft, W.B., Levine, B., Lesser, V. A Multi-Agent Approach for Peer-to-Peer-based Information Retrieval Systems. In In *Proc. of AAMAS'*04, 456-464.

[19] Zhang, H., Lesser, V. Multi-Agent Based Peer-to-Peer Information Retrieval Systems with Concurrent Search Sessions. In *Proc. of AAMAS'*06, 305-31.