

# A Self-Organizing Ant-based Information Gossiping Algorithm for P2P Networks

Christophe Guéret, Nicolas Monmarché, Mohamed Slimane

Université François Rabelais Tours, Laboratoire d'Informatique,  
Polytech'Tours, 64 avenue Jean Portalis - 37200 Tours, France  
{gueret,monmarche,slimane}@univ-tours.fr

**Abstract:** They appeared in our life only few years ago and now they are everywhere: computers have become ubiquitous and, almost, irreplaceable. Classical ways of creating, managing and exchanging information have been progressively replaced by electronic means. Everyday, information diffusion tools like the World Wide Web, E-mails, Forums and other Blog software are now commonly used. However, in spite of this plebiscite, computer based information managing still suffers some weaknesses. Mainly, software aimed to do CSCW (Computer Supported Collaborative Work) can be blamed for requiring the user to do an effort to use them. In this paper we present an algorithm aimed at perform autonomous selective dissemination of messages within a network. It constitutes the communication layer of our framework called PIAF ("Personal Intelligent Agent Framework") which is intended to help users transparently share information. This algorithm works in a fully decentralized way, using epidemic diffusion mechanism and artificial ants paradigm to achieve self-organization and information flows management.

## 1 Introduction

Computer based technology occupy an important place in our daily life and now are considered to be ubiquitous [WGB99]. During the last decades, using computers has modified users' habits. Electronic documents have changed the way to write, archive and diffuse content while Internet has changed the way we collaborate. Now, it is possible to work on a same project, exchanging documents or chatting regardless the physical positions of the co-workers.

People doing collaborative work may have to share two types of content: data and knowledge. Data can be any electronic container such as text files, video or web links. Emails and file-sharing software are among the most used form of data sharing. Knowledge is related to the user's mind and may not be represented as a data file on a computer. For instance, a recommendation about a good restaurant or an advice concerning a research topic are both examples of knowledge sharing. Supposing it is possible to represent knowledge as a data file to share, from now on, we will use the generic term of "resource" to design both shared data and knowledge.

Depending on the user, we can distinguish between two kinds of resource sharing: implicit

or explicit. Sending an email is an explicit act while using a software to share idle CPU time is implicit for the user. Explicit sharing is the most challenging task for the user. Let us suppose a user finds an interesting website and wants to have all other, potentially interested, peers know about it. The strategy may be either 1) within the set of known peers, inform the subset of peers more likely to be interested by the website 2) inform all peers and let them decide if they are interested or not. In the first case, the risk for the sender is to omit some interested peers while in the second, the risk is to spam (*ie*: sending unwanted messages to some peers). This problem is one of the difficulties related to the usage of Computer Supported Collaborative Work (CSCW) software [Gru88]:

- Lack of mutual awareness: sharing content efficiently implies a global knowledge of the peers' needs. In order not to bother every single one, every user should know what his/her peers are interested in. But users may randomly appear and disappear in the network. Also they may be interested in different domains or make spurious searches from time to time. Thus, maintaining such knowledge is difficult.
- Users might not be motivated enough in using a software helping them sharing resources. Such software may involve, for instance, sending emails to people inside the network or using dedicated tool to tell them about what they have found. In both cases the user has to make an effort. Users generally do not like to change their habits and such solutions may weaken their motivation and dissuade them from using the software.
- Users can not define precisely what they are interested in: if we take the example of web browsing, users are most likely to jump from page to page looking for interesting links rather than follow a precise and predetermined path.

We believe that a resource sharing system based on implicit sharing could cope with those problems. We have designed the PIAF software in order to follow this idea. PIAF stands for Personal Intelligent Agent Framework. This framework is divided in two main layers: communication and dialog. The communication layer takes in charge information flows within the network. The dialog layer is the interface between the user and the network. In this paper we focus on the communication layer.

Our algorithm uses an ant paradigm, an idea which has already been explored in the context of content-based searches in unstructured P2P networks [BMM02]. Many ant species are known to use chemical trails to perform navigation and food exploitation. In this case, the involved volatile substances, called pheromones, are a kind of indirect communication mean between workers of the colony. These biological principles have been translated to many combinatorial problems modeled with a graph [BDT99] and even when a network routing problem are considered (see for instance the case of mobile ad hoc networks [DDG05]). The good behavior of this class of algorithm, especially within distributed environment, has led us to use similar principles.

The reminder of this paper is organized as follows. In section 2 we discuss some existing solutions for message circulation and dynamic topology management in P2P networks and state on the originality of our proposition. The following section 3 presents the algorithms

we have developed. Finally, we present experimental results in sections 4 and 5 before concluding in section 6.

## 2 Information flows and rewiring schemes

Message flows in a network are generated by exchanges between a server having a shared resource and a client asking for it. Actually, in a network of collaborators each user may act as a client (looking for a resource) or a server (informing users about what it shares). We are then in front of a so-called P2P network. Because of this duality of roles, finding a given resource is not easy. The tricky task for a client being mainly to find a relevant provider. In centralized networks such as Napster [Nap03], a server knowing which resources are shared and who is sharing them in the network is used to find them. However, those solutions have proved not to scale efficiently and fully decentralized P2P architectures are preferred. In this context, two main strategies can be considered for resources exchanges: they correspond to the "push" and "pull" strategies for information exchange as introduced by [FZ98]:

- *Query processing*: A users sends a query and the system returns a list of peers to contact. In a structured overlay, the location of peers and data depends on their respective identifiers. Usually, the search space is partitioned among peers with a Distributed Hash Table [IMK<sup>+</sup>01] algorithm or a hierarchical ordering of identifiers [PRR97]. A query is routed from peer to peer until it reaches a peer having a particular identifier. On the opposite, an unstructured overlay does not rely on a pre-defined architecture. A query is blindly (*e.g.* without information on the underlying topology) spread in the network until a result is obtained.
- *Selective dissemination*: The dual scenario of query processing is when users do not send query at all. Instead, an event based system is considered: clients subscribe to the event service by submitting a profile while servers publish events which will be dispatched to attended recipients. The publish/subscribe (pub-sub for short) scheme may be topic-based or content-based depending if the profile defines constraints concerning the topic of the event or its content. If the pub-sub is built over a structured overlay, events are routed. Whereas over an unstructured overlay, probabilistic dissemination (*e.g.* Gossip) is a more suitable strategy. Gossip protocols [DGH<sup>+</sup>88] consist in sending a message to a subset of the connected peers, according to a given probability.

The communication algorithm of PIAF is aimed at automatically disseminating news using a gossip-based diffusion scheme. Hence it falls under the category of selective dissemination algorithms. The peers will have to be informed of the existence and location of shared resources. Following the idea of Newscast [VJvS03], we will refer to this information as a "News". The objective is not to have a reliable multicast, that is which ensures that all peers recover all existing news [DGH<sup>+</sup>88, CAPMN03, VJvS03], but, instead, which performs a directed and focused diffusion [LM00]. A given information is gossiped to peers

more likely to be interested in it. Unlike other selective information dissemination systems for P2P network [KTID03], we do not consider the user has to define a profile to get useful news. Our approach is similar to the concept of autonomous gossiping introduced by Datta *et al.* [DQA04] although, unlike them, we do not use individual profiles to defines users interests nor we associate categories to news.

Each peer is connected to a limited amount of other peers. This provides them with a partial view of the whole network. Dynamic topology is used to adapt this set of connections according to a given criteria. It has been observed that a social network of collaborator exhibits small world properties: the network is made of many dense groups loosely connected to each other [WS98]. Those groups appear when individuals congregate as they found themselves having shared center of interest. Dynamically adjusting the topology of the P2P overlay network in order to make it similar to the underlying small world can improve sharing efficiency. A criteria is used to decide if two peers have similar interests or not. Depending on it, a given connection may be dropped or kept. To compute this criteria, it is necessary to have a model of peer's interests. This model, usually referenced to as a profile, may commonly be published by the peers [HS04] or exchanged on demand [Sch04]. We propose a third new strategy inspired by the ideas of overhearing [BSSZ01] and use of information trails [Pay98] in a network. We consider that whenever a peer sends a message over the network, he gives an hint about what he is interested into. Hence, instead of inquiring about the expertises of one's peers, we guess them from traffic they generate over the network.

### 3 PIAF Communication layer

One can view the P2P network as a directed graph where each node  $n_i \in \mathcal{N}$  is a peer. An edge  $(i, j) \in \mathcal{C}(t)$ , represents a connection from a peer  $n_i$  to a peer  $n_j$  present at an instant  $t$ . Introducing the set of possible arc in the graph  $\mathcal{E}$ , we have  $\mathcal{C}(t) \subset \mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ . For a given peer  $n_i$ , we define the neighborhood  $V_i(t) = \{n_j \in \mathcal{N} \mid (i, j) \in \mathcal{C}(t)\}$  has the subset of peers it is connected to. This model defines a social network, that is a network in which edges define relations between nodes. In our case, the relations reflect shared centers of interest. As stated earlier, groups tend to appear in social networks. The density of those groups is measured by a clustering coefficient  $\gamma_i(t)$ . For a peer  $n_i$ , it quantifies how dense its neighborhood is. Therefore, if  $\gamma_i(t) \simeq 1$ , then  $n_i$  is considered to be part of a dense group.

$$\gamma_i(t) = \frac{\text{total edges in } V_i(t)}{\text{total possible edges in } V_i(t)} \quad (1)$$

Artificial pheromones are defined on a vector space  $\mathbb{R}^n$ . Each news item  $I$  has an associated pheromone vector  $\tau(I)$ . Pheromone vectors  $\tau_{i \rightarrow j}(t)$  are also associated to connections  $(i, j)$ . We suppose the existence of a similarity  $s$  defined on this vector space  $s : \mathbb{R}^n \times \mathbb{R}^n \mapsto [0; 1]$ . Thus, it is possible to evaluate the similarity between two connections, as well as the similarity of a connection related to a news item. On a connection, pheromones are used as a memory for news received from other peers. Therefore, a given

peer  $n_i$  will store incoming pheromone vectors  $\tau_{i \leftarrow j}(t), j \in V_i(t)$  and update them every time he receives a news from one of its neighbors.

### 3.1 Ant's gossiping activity

Ants work as follows when disseminating a news item  $I$ . Every  $T_g$  unit of time, the ant will try to push  $I$  from its nest  $n_i$  to another nest  $n_j$  randomly chosen in  $V_i(t)$ . This activity consists in first choosing a destination and then update pheromones. Ants stop the diffusion of  $I$  once it has decided to stay at the nest  $k$  consecutive times. This parameter denotes its patience.

#### 3.1.1 Choose a destination

A stochastic algorithm is used to select a destination  $n_j$  within  $V_i(t)$ . According to a similarity threshold  $s_{min}$ , neighbors are first sorted in two groups depending if they are likely to be interested by  $I$  or not. This similarity estimates if  $I$  is similar to other news items previously sent to a given neighbor. Those groups are respectively defined as  $V_i(I, t)$  and  $\overline{V_i(I, t)}$ .

$$V_i(I, t) = \{n_j \in V_i(t) \mid s(\tau_{i \leftarrow j}(t), \tau(I)) \geq s_{min}\} \quad (2)$$

$$\overline{V_i(I, t)} = \{n_j \in V_i(t) \mid n_j \notin V_i(I, t)\} \quad (3)$$

This classification is also used to update two counters  $PV_{i \rightarrow j}(t)$  and  $\overline{PV_{i \rightarrow j}(t)}$  used to record how many positive valuations (PV) a given connection received. For a neighbor  $n_j$ ,  $PV_{i \rightarrow j}(t)$  is incremented if  $n_j \in V_i(I, t)$  whereas  $\overline{PV_{i \rightarrow j}(t)}$  is incremented if  $n_j \in \overline{V_i(I, t)}$ . Until it has finished diffusing it, an ant is not allowed to send a news item twice to a same peer. Thus a subset of  $V_i(I, t)$  and  $\overline{V_i(I, t)}$  is defined where visited peers  $seen(I, t)$  are excluded.

$$V_i^{unseen}(I, t) = V_i(I, t) \setminus seen(I, t) \quad (4)$$

$$\overline{V_i^{unseen}(I, t)} = \overline{V_i(I, t)} \setminus seen(I, t) \quad (5)$$

The probability  $P_{i \rightarrow j}(I, t)$  of a peer to be elected as a destination by the ant depends of the group it was assigned to (see equation (6)). For an interested peer, this probability is proportional to its relative similarity with  $I$ . Meanwhile, non interested peers may be equiproportionally chosen. Sending to either an interested or not interested peer is a matter of exploitation versus exploration considering the problem of finding an optimal messages flow. Performing only exploitation can help reaching this optimum but, on the other hand, exploration is needed to find news peers to connect to. Therefore, a trade-off must be found to allow trying to send news items to some other neighbors even if they does not seem to be interested. To achieve this, ants are granted with a notion of "freewill". According to a

probability  $\eta$ , an ant may choose to select a destination likely to be interested or not. Also, it has  $n^+$  times more chances to stay at the nest rather than sending  $I$  to a non interested peer.

$$P_{i \rightarrow j}(I, t) = \begin{cases} \frac{\eta \cdot \delta(|V_i(I, t)| > 0) \cdot s(\tau_{i \leftarrow j}(t), \tau(I))}{\sum_{z \in V_i(I, t)} s(\tau_{i \leftarrow z}(t), \tau(I))} & \text{if } n_j \in V_i^{unseen}(I, t), \\ \frac{1 - \eta \cdot \delta(|V_i(I, t)| > 0)}{|V_i^{unseen}(I, t)| + n^+} & \text{if } n_j \in \overline{V_i^{unseen}(I, t)} \\ \frac{n^+ \cdot (1 - \eta \cdot \delta(|V_i(I, t)| > 0))}{|V_i^{unseen}(I, t)| + n^+} & \text{if } i = j \end{cases} \quad (6)$$

### 3.1.2 Adjust pheromones

This step on the algorithm only occurs when the ant decides not to stay in the nest. On its way toward the peer  $n_j$  it has chosen, the ant will lay down pheromones. The amount of pheromones is defined by a factor  $\rho(I, \Delta t)$ , both used for evaporation and deposit of pheromones.

$$\tau_{j \leftarrow i}(t + \Delta t) = (1 - \rho(I, \Delta t)) \cdot \tau_{j \leftarrow i}(t) + \rho(I, \Delta t) \cdot \tau(I) \quad (7)$$

with  $\Delta t$  the time elapsed since a message was last transferred through this connection.  $\rho(I, \Delta t)$  depends on two factors: the activity on the link and the source of  $I$ . The more messages are transferred through a connection, the more pheromones deposit will be important. Also, pheromones deposit should decrease as the news item is farther away from its origin. We have chosen to use a Gaussian for each factor and defined a maximum amount of deposit  $\rho_{max}$  (see equation 8).

$$\rho(I, \Delta t) = \rho_{max} \exp^{-\alpha r(I)} \exp^{-\left(\frac{\Delta t}{\sigma}\right)^2} = \rho_{max} \exp^{-\left(\frac{\Delta t}{\sigma}\right)^2 - \alpha r(I)} \quad (8)$$

$\alpha$  and  $\sigma$  are two regulation factors used to adjust the trade-off wanted between reactivity and memory of the system.  $r(I)$  is the round count, that is, the number of peers this news item has crossed by since its creation.

### 3.2 Nest mobility

Moving the nest consists in modifying its neighborhood by adding or removing some links. A peer may have a maximum of  $V_{max}$  opened connections. To move the nest, the first step is to try fetching a contact from a directory of peers  $n_i$  knows. In case of success, this peer is contacted, otherwise,  $n_i$  ask one of his neighbor for help.

### 3.2.1 Connect to a new peer

Supposing such a contact is found, and before connecting to him, the peer still has to verify if  $|V_i(t)| < V_{max}$ . If not, the less efficient connection would be dropped.

1. Efficiency is defined as a ratio between the number of time a peer was estimated to be interested and the total number of estimations performed by ants.

$$\forall n_j \in V_i(t), U_j(t) = \frac{PV_{i \rightarrow j}(t)}{PV_{i \rightarrow j}(t) + \overline{PV_{i \rightarrow j}(t)}} \quad (9)$$

2. If  $U(n_j)$  falls under a given threshold  $\beta$ , the connection is not considered not to be efficient enough. Hence it has a probability  $P_{i \rightarrow j}^{drop}(t)$  to be dropped. The lower  $U_j(t)$ , the higher this probability is.

$$\forall n_j \in V'_i(t), P_{i \rightarrow j}^{drop}(t) = \frac{\beta - U_j(t)}{\sum_{z \in V'_i(t)} \beta - U_z(t)} \quad (10)$$

with  $V'_i(t) = \{n_j \in V_i(t) | U(n_j) < \beta\}$  the subset of inefficient neighbors.

3. If  $|V_i(t)| < V_{max}$  a connection is established with the peer previously picked up from the directory. Otherwise, all connections were useful and no one was dropped.

### 3.2.2 Ask for a suggestion

If  $n_i$  was not able to find a peer in its directory, it asks one of its neighbors to send him a suggestion. It sends to a peer  $n_j$  a message with a copy of its directory.  $n_j$  then browses its own directory and answers to  $n_i$  sending him back the address of the peer most likely to be useful for him. The peer  $n_j$  is itself picked from  $V_i(t)$  with a rank based selection based on similarities.

## 4 Simulation environment

A discrete event simulator was used in order to implement and test our information dissemination and topology management algorithms. We have chosen to use the OmnetPP discrete event simulator [Var02].

We make no assumption concerning the nature of metadata represented by the pheromones. During all the algorithms steps, only the similarity between two vectors is considered. Hence, we choose to generate an artificial dataset that suits to our needs. It is made of 4 categories of news items  $C_1, C_2, C_3$  and  $C_4$ , each populated with 100 vectors of dimension  $n = 100$ . For the simulations, the similarity used is the standard cosine (equation

11). The definition of the topics is not directly used in the algorithms, it is only used to compute the values of performance criteria.

$$\forall A, B \in \mathbb{R}^n, s(A, B) = \frac{\sum_{i=1}^n a_i * b_i}{\sqrt{\sum_{i=1}^n a_i^2} \sqrt{\sum_{i=1}^n b_i^2}} \quad (11)$$

The average similarity between two elements from two distinct classes is, at maximum, of 0.25 and, at minimum, of 0.08. The average similarity for two news items from the same class is 0.74. Ant's  $\lambda$  parameter is set to 0.7 in order to have high probability to correctly recognize elements of a same class.

Each peer is only allowed to connect to 4 other peers. Initially, no connection is established and in their directory peers have the address of a unique randomly chosen peer. To simulate the presence of a user, each peer as an agent periodically sends news items related to the peer's center of interest. News items are grabbed from a global repository ensuring no same news is send twice by two different peers. For the simulations, the generation period was set to  $\mathcal{U}[100, 1500]$  units of time and the amount of news produced was limited to 10 items.

The performance of the diffusion algorithm is evaluated through 3 estimators: the network clustering coefficient  $\gamma$  and averaged values for completeness and efficiency factors. Network clustering coefficient  $\gamma$  is the average of clustering coefficient for each peer as defined in equation 1. Completeness defines the number of interesting news item a peer gathered compared to the total amount of interesting news items available in the network. Precision is the ratio between the number of interesting news divided by the total number of news fetched. Those definitions are similar to classical recall and precision but, in the our context, does not have the same meaning.

## 5 Results

Because of space constraints, the results presented here deal with the adjustment of  $\beta$ . The objective is to test when it is worth considering a peer being inefficient.

### 5.1 First test

In this first test, the network is made of 20 peers interested by one of the four subject previously defined. Five peers are assigned to each of the topics of the dataset. They are supposed to be only interested in the topic they were assigned to. Figure 1 shows the evolution of clustering coefficient for different values of  $\beta$ .

The best result is obtained with  $\beta = 0.2$ . Higher values of  $\beta$  lead to drop many connections, therefore almost no cluster can be formed. On the other hand, if  $\beta$  is set to lower values, fewer connections may be dropped and the network becomes static. The figure 2 shows how information dissemination evolves during the simulation. Worst results are for



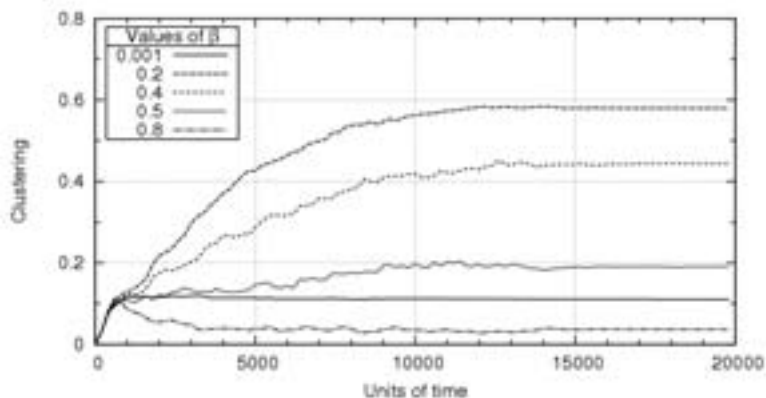


Figure 1: Evolution of clustering coefficient for various values of  $\beta$

$\beta = 0.8$ , when the network can not be clustered. For 0.2 and 0.4, completeness is similar but precision differs. This shows that a peer will get the same amount of interesting news in the two cases but if  $\beta = 0.4$  it will also get more less interesting news items.

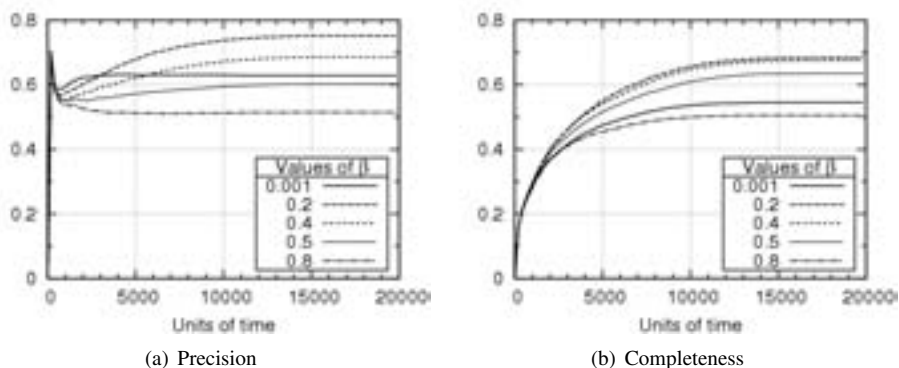


Figure 2: Evolution of completeness and precision for various values of  $\beta$

Considering the small amount of news produced and the frequency of publication, the initialization phase does not last longer than 15000 units of time. Hence, the figures depict the stabilization of the system when no more news item are injected. The system is tested with 2 ants by peer, each having a patience  $k$  of 2. Best results are achieved when  $PV_{i \rightarrow j}(t) > 4 \cdot PV_{i \rightarrow j}(t)$ , that is when the two ants had evaluated twice a connection has being useless. This tends to prove that the influence  $\beta$  is related to  $k$  and the number of ants.

## 5.2 Stability of result

In order to check the stability of this result, we have performed other tests using a different dataset and networks of different sizes. This second test involve 4 networks of 20, 25, 40 and 100 peers interested only in one subject among 4 or 5 available topics. Figure 3 shows the final, stabilized, value of clustering coefficient. For each network, this final value has been averaged over 50 runs of the algorithm.

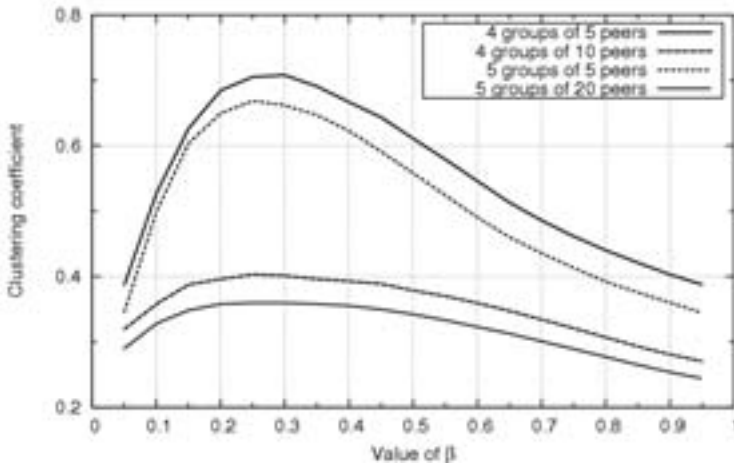


Figure 3: Evolution of clustering coefficient for various values of  $\beta$

The first interesting result is that, independently from the size of the network, a maximum value for the clustering coefficient is reached when  $\beta = 0.25$ . That confirms the conclusion of the first presented test. An other constant result is the profile of the curves drawn which proves that the behavior of the rewiring algorithm is not influenced by the size of the network.

## 6 Conclusion and future directions

In this paper, we have presented an algorithm using artificial ants to diffuse information in a P2P network. The information diffusion is proactive and transparent for the user. Our main contribution is the use of estimated profiles in order to perform probabilistic broadcast. We use the artificial ants paradigm where artificial pheromones defines a memory for information exchanged. Firsts tests has led to an earlier version of the algorithm and proved the interest of using estimated profiles [GMS05, GMS06]. In this paper, we highlight a relation between the number of ants, their patience and the tolerance in estimating the utility of neighbors. During future development, tests on larger datasets along with a theoretical study will be performed in order to confirm this tendency. Also, we consider developing third party applications mandatory to replace the agent creating news by a real

user and perform real life tests.

## References

- [BDT99] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, New York, 1999.
- [BMM02] Ozalp Babaoglu, Hein Meling, and Alberto Montresor. Anthill: A Framework for the Development of Agent-Based Peer-to-Peer Systems. In *Proceedings of the 22th International Conference on Distributed Computing Systems (ICDCS '02)*, Vienna, Austria, July 2002.
- [BSSZ01] P. Busetta, L. Serafini, D. Singh, and F. Zini. Extending Multi-Agent Cooperation by Overhearing. Technical Report 0101-01, Istituto Trentino di Cultura, January 2001.
- [CAPMN03] F. M. Cuenca-Acuna, C. Peery, R. P. Martin, and T. D. Nguyen. PlanetP: Using Gossiping to Build Content Addressable Peer-to-Peer Information Sharing Communities. In *Proceedings of the 12th International Symposium on High Performance Distributed Computing (HPDC)*, June 2003.
- [DDG05] Frederick Ducatelle, Gianni Di Caro, and Luca Maria Gambardella. Using ant agents to combine reactive and proactive strategies for routing in mobile ad hoc networks. *International Journal of Computational Intelligence and Applications (IJCIA)*, 5(2):169–184, 2005. Special Issue on Nature-Inspired Approaches to Networks and Telecommunications.
- [DGH<sup>+</sup>88] Alan J. Demers, Daniel H. Greene, Carl Hauser, Wes Irish, John Larson, Scott Shenker, Howard E. Sturgis, Daniel C. Swinehart, and Douglas B. Terry. Epidemic Algorithms for Replicated Database Maintenance. *Operating Systems Review*, 22(1):8–32, 1988.
- [DQA04] Anwitaman Datta, Silvia Quarteroni, and Karl Aberer. Autonomous Gossiping: A Self-Organizing Epidemic Algorithm for Selective Information Dissemination in Wireless Mobile Ad-Hoc Networks. *LNCS*, 3226:126–143, 2004.
- [FZ98] M.J. Franklin and S.B. Zdonik. "Data in Your Face" : Push Technology in Perspective. In *Proceedings ACM SIGMOD International Conference on Management of Data*, pages 516–519, 1998.
- [GMS05] C. Guéret, N. Monmarché, and M. Slimane. Aide à la navigation sur Internet : utilisation de fourmis artificielles pour l'échange d'informations dans un réseau P2P. In *Actes du congrès de la ROADEF*, 2005.
- [GMS06] Christophe Guéret, Nicolas Monmarché, and Mohamed Slimane. Sharing Resources with Artificial Ants. In *Proceedings of the 9th International Workshop on Nature Inspired Distributed Computing (NIDISC'06)*, Rhodes Island, Greece, April, 25-29 2006. 8 pages CD-ROM.
- [Gru88] Jonathan Grudin. Why CSCW applications fail: problems in the design and evaluation of organization of organizational interfaces. In *Proceedings of the 1988 ACM conference on Computer-supported cooperative work*, pages 85–93, Portland, Oregon, United States, 1988.

- [HS04] Peter Haase and Ronny Siebes. Peer selection in peer-to-peer networks with semantic topologies. In *Proceedings of the 13th International World Wide Web Conference*, New York City, NY, USA, 2004.
- [IMK<sup>+</sup>01] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoe, and Hari Balakrishnan. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. In *SIGCOMM'01*, August 27-31 2001.
- [KTID03] Manolis Koubarakis, Christos Tryfonopoulos, Stratos Idreos, and Yannis Drougas. Selective Information Dissemination in P2P Networks: Problems and Solutions. *SIGMOD Record, Special Issue on Peer-to-Peer Data Management*, 32(3):71–76, 2003.
- [LM00] Meng-Jang Lin and Keith Marzullo. Directional Gossip: Gossip in a Wide Area Network. In *Proceedings of European Dependable Computing Conference*, 2000.
- [Nap03] Napster. Napster file sharing. <http://www.napster.com>, 2003.
- [Pay98] David W. Payton. Discovering Collaborators by Analyzing Trails Through an Information Space. In *AAAI Fall Symposium on Artificial Intelligence and Link Analysis*, October 23-25 1998.
- [PRR97] C. Greg Plaxton, Rajmohan Rajaraman, and Andrea W. Richa. Accessing Nearby Copies of Replicated Objects in a Distributed Environment. In *Proceedings of ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, June 1997.
- [Sch04] Christoph Schmitz. Self-Organization of a Small World by Topic. In *First International Workshop on Peer-to-Peer Knowledge Management (P2PKM)*, August 2004.
- [Var02] András Varga. OMNeT++ discrete event simulation environment. [www.omnetpp.org](http://www.omnetpp.org), 2002.
- [VJvS03] Spyros Voulgaris, Mark Jelasity, and Maarten van Steen. A Robust and Scalable Peer-to-Peer Gossiping Protocol. In *Proceedings of the 2nd International Workshop on Agents and Peer-to-Peer Computing (AP2PC03)*, Melbourne, Australia, 2003.
- [WGB99] M. Weiser, R. Gold, and J. S. Brown. The origins of ubiquitous computing research at PARC since the late 1980s. *IBM Systems Journal*, 38(4):693–696, 1999.
- [WS98] Duncan J. Watt and H. Strogatz Steve. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 4 June 1998.