A self-organizing ant-based information gossiping algorithm for P2P networks

Christophe Guéret¹, Nicolas Monmarché¹, and Mohamed Slimane¹

Université François-Rabelais de 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. In spite of this plebiscite, computer based information managing still suffers some weaknesses. Compared with communications between attendees of a meeting, information exchanges over the Internet can appear to be very poor. Mainly, software aimed to do CSCW (Computer Supported Collaborative Work) can be blamed for requiring the user to do an effort to use them. Also, being able to communicate with a lot of people does not really ease the task of recipients selection for a user willing to share information. 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.
A self-organizing ant-based information gossiping algorithm for P2P networks

Christophe Guéret¹, Nicolas Monmarché¹, and Mohamed Slimane¹

Université François-Rabelais de 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. In spite of this plebiscite, computer based information managing still suffers some weaknesses. Compared with communications between attendees of a meeting, information exchanges over the Internet can appear to be very poor. Mainly, software aimed to do CSCW (Computer Supported Collaborative Work) can be blamed for requiring the user to do an effort to use them. Also, being able to communicate with a lot of people does not really ease the task of recipients selection for a user willing to share information. 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 [1]. During the last decades, using computers 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’s possible to work on a same project, exchanging documents or chatting regardless the physicals 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 no be represented as a data file on a computer. For instance, a recommendation about a good restaurant or an advise 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 [2]:

- 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 a resource sharing system based on implicit sharing could cope with those problems. The PIAF software we design is based on 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 the 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 as already been explored in the context of content-based searches in unstructured P2P networks [3].

The reminder of this paper is organized as follows. In section 2 we discuss some existing solutions for message circulation in P2P networks and dynamic topology management and state on the originality of our proposition. The following section 3 presents the algorithms we developed. Finally, we present experimental results in sections 4 and 5 and conclude 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 an resource) or a server (informing users about what it shares). We are then in presence 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 [4], a server knowing which resources are shared and who shares them in the network is used to find them. However, those solutions have proved not to scale and fully decentralized P2P architectures are preferred.

In this context, two main strategies can be considered for resources exchanges:

**Query processing**: A users express a query and the systems returns a list of peers to contact. In a structured overlay, the placement of peers and data depends on their respective identifiers. Usually, the search space in partitionned among peers with a Distributed Hash Table [5] algorithm or a hierarchical ordering of identifiers [6]. 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 relies on a predefined architecture. A query is blindly (e.g. without information on the underlying topology) diffused in the network until a result is found.

**Selective dissemination**: The dual scenario of query processing is when users sends no query at all. Instead, an event based system is considered: clients subscribes 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 contraints 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 [7] consists in sending a message to a subset of the connected peers, according to a given probability.

Those two schemes corresponds to the "push" and "pull" strategies for information exchange as introduced by [8].

In selective dissemination peers are informed of the existence and location of shared resources. Following the idea of Newscast [9], we will refer to this information as a "News". The communication algorithm of PIAF is aimed at automatically disseminating news using a gossip-based diffusion scheme. The objective is not to have a reliable multicast, that is, ensuring all peers recover all existing news [7,10,9] but, instead, performing a directed and focused diffusion [11]. A given information is gossiped to peers more likely to be interested in it. Unlike other selective information dissemination systems for P2P network [12], we do not consider the user has to define a profile to get usefull news. Our approach is similar to the concept of autonomous gossiping introduced by Datta et al. [13] 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 [14]. 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 [15] or exchanged on demand [16]. We propose a third new strategy inspired by the ideas of overhearing [17] and use of information trails [18] 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 expertise 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 were each node $n_i \in \mathcal{N}$ is a peer. An edge $(i, j) \in C(t)$, 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 $C(t) \subseteq \mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$. For a given peer $n_i$, we define the neighborhood $V_i(t) = \{n_j \in \mathcal{N} \mid (i, j) \in C(t)\}$ has the subset of peers it is connected to. This model defines a social network, that is a network which edges defines relations between nodes. In our case, the relations reflects shared centers of interest. As stated earlier, groups tends to apper in social networks. The density of those groups is measured by a clustering coefficient $\gamma_i(t)$ which, for a peer $n_i$, quantifies how dense its neighborhood is. If $\gamma_i(t) \simeq 1$, $n_i$ 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 pheromones vector $\tau(I)$ and vectors $\tau_{i \rightarrow j}(t)$ are associated to connections $(i, j)$. We suppose the existence of a similarity $s$ defined on this space vector $s : \mathbb{R}^n \times \mathbb{R}^n \mapsto [0; 1]$. 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 pheromones vectors $\tau_{i \rightarrow j}(t), j \in V_i(t)$ and update them every time he receives a news from one of its neighbors.

3.1 Ant’s gossiping activity

Ants works as follows when disseminating a news item $I$. Every $T_g$ unit of time, the ant will push this news 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.

**Choose a destination**

A stochastic algorithm is used to select \( n_j \) among \( V_i(t) \). According to a similarity threshold \( s_{\text{min}} \), neighbors are first sorted in two groups depending if they are likely to be interested by \( I \) or not. Those groups are respectively defined has \( V_i(I, t) \) and \( \overline{V}_i(I, t) \).

\[
V_i(I, t) = \{ n_j \in V_i(t) \mid s(\tau_{i\rightarrow j}(t), \tau(I)) \geq s_{\text{min}} \} \tag{2}
\]

\[
\overline{V}_i(I, t) = \{ n_j \in V_i(t) \mid n_j \notin V_i(I, t) \} \tag{3}
\]

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 positives 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.

\[
V_{i}^{\text{unseen}}(I, t) = V_i(I, t) \setminus \text{seen}(I, t) \tag{4}
\]

\[
\overline{V}_{i}^{\text{unseen}}(I, t) = \overline{V}_i(I, t) \setminus \text{seen}(I, t) \tag{5}
\]

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 mater of exploitation versus exploration. Exploitation can help having an optimal messages flow 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 have 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.

\[
P_{i\rightarrow j}(I, t) = \begin{cases} 
\frac{\eta \cdot \delta(|V_i(I, t)| > 0) \cdot s(\tau_{i\rightarrow j}(t), \tau(I))}{\sum_{z \in V_i(I, t)} s(\tau_{i\rightarrow z}(t), \tau(I))} & \text{if } n_j \in V_{i}^{\text{unseen}}(I, t), \\
\frac{1 - \eta \cdot \delta(|V_i(I, t)| > 0)}{|V_{i}^{\text{unseen}}(I, t)| + n^+} & \text{if } n_j \in \overline{V}_{i}^{\text{unseen}}(I, t), \\
\frac{n^+ \cdot (1 - \eta \cdot \delta(|V_i(I, t)| > 0))}{|V_{i}^{\text{unseen}}(I, t)| + n^+} & \text{if } i = j
\end{cases} \tag{6}
\]
Adjust pheromones

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

$$\tau_{j \rightarrow i}(t + \Delta t) = (1 - \rho(I, \Delta t)) \cdot \tau_{j \rightarrow i}(t) + \rho(I, \Delta t) \cdot \tau(I)$$  \hspace{1cm} (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 source. We chose to use a gaussian for each factor and defined a maximum amount of deposit $\rho_{max}$ (see equation 8).

$$\rho(I, \Delta t) = \rho_{max} \exp^{-\alpha r(I)} \exp^{-(\frac{\Delta t}{\gamma})^2} = \rho_{max} \exp^{-(\frac{\Delta t}{\gamma})^2 - \alpha r(I)}$$  \hspace{1cm} (8)

$\alpha$ and $\gamma$ are two regulation factors used to adjust the trade-off wanted between reactivness and memory of the system. $r(I)$ is the number of peers this news as crossed by since its creation. Ants stop the diffusion of $I$ once it has decided to stay at the nest $k$ consecutive times. This parameter denotes its patience.

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.

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.

$$\forall n_j \in V_i(t) \hspace{0.2cm} U_j(t) = \frac{PV_{i \rightarrow j}(t)}{PV_{i \rightarrow j}(t) + PV_{i \rightarrow j}(t)}$$  \hspace{1cm} (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.

$$\forall n_j \in V'_i(t) \hspace{0.2cm} P_{i \rightarrow j}^{drop}(t) = \frac{\beta - U_j(t)}{\sum_{z \in V'_i(t)} \beta - U_z(t)}$$  \hspace{1cm} (10)

with $V'_i(t) = \{ n_j \in V_i(t) | U(n_j) < \beta \}$ the subset of inefficient neighbors.
3. If $|V_i(t)| < V_{\text{max}}$ a connection is established with the peer previously picked up from the directory. Otherwise, all connections were useful and no one was dropped.

**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 $n_j$ 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 algorithm. We have chosen to use the OmnetPP discrete event simulator [19].

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 our needs. It is made of 4 categories of informations $C_1, C_2, C_3$ 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.

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

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 informations 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.

The network is made of 20 peers interested by one of the four subject previously defined. Each peer is only allowed to connect to 4 other peers. 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. 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 informations 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 $U[100, 1500]$ 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 contraints, the result presented here deals with the adjustment of $\beta$. The objective is to test when it is worth considering a peer being unefficient. Figure 1 shows the evolution of clustering coefficient for different values of $\beta$.

![Fig. 1. Evolution of clustering coefficient for different values of $\beta$.](image)

The best result is obtained with $\beta = 0.2$. Higher values of $\beta$ leads 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. Worsts results are for $\beta = 0.8$, 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.

Considering the small amount of news produced and the frequency of publication, the initialisation phase does not last longer than 15000 units of time. Hence, the figures depict the stabilisation 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. Bests 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.
6 Conclusion and future directions

We presented in this paper 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 an artificial ants paradigm where artificial pheromones defines a memory for information exchanged. Firsts tests led on an earlier version of the algorithms proved the interest of using estimated profiles [20, 21]. 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 therotical study will be performed in order to confirm this further. Also, we consider developping third party applications mandatory to replace the agent creating news by a real user and perform real case tests.

References


