Heterogenous dating service with application to rumor spreading

Olivier Beaumont, Philippe Duchon, Miroslaw Korzeniowski

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Système d'organisation de rendez-vous hétérogène et application à la diffusion de rumeurs

Olivier Beaumont
LaBRI – INRIA Futurs
obeaumon@labri.fr

Philippe Duchon
LaBRI – Univ. Bordeaux
duchon@labri.fr

Miroslaw Korzeniowski
LaBRI – INRIA Futurs

korzenio@labri.fr

N° 6168

Avril 2007

Thème COM

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--6168--FR+ENG

Systme d'organisation de rendez-vous htrogne et application la diffusion de rumeurs

Olivier Beaumont
LaBRI – INRIA Futurs
obeaumon@labri.fr

Philippe Duchon
LaBRI – Univ. Bordeaux
duchon@labri.fr

Miroslaw Korzeniowski
LaBRI – INRIA Futurs
korzenio@labri.fr

Thème COM — Systèmes communicants
Projet Cépage

Rapport de recherche n° 6168 — Avril 2007 — 15 pages

Résumé : Dans cet article, nous décrivons un mécanisme compltement distribué, appelé "dating service", pour organiser des communications dans un réseau hétérogne, tout en assurant que les capacités de communication des nœuds ne sont pas excédées. Nous montrons qu'avec forte probabilité, ce mécanisme assure qu'une fraction constante des communications possibles est effectivement organisée. Plus intéressant encore, nous montrons également que cette propriété demeure même si les nœuds n'ont pas la possibilité de choisir un autre nœud de manière aléatoire uniforme. Nous décrivons également, comme application du "dating service" un algorithme pour la diffusion de rumeurs qui permet de diffuser un message de taille unitaire l'ensemble des nœuds d'une plate-forme pair--pair en un nombre logarithmique d'étapes, avec forte probabilité.

Mots-clés : algorithmes randomisés, algorithmes distribués, diffusion de rumeurs, plates-formes hétérognes

Unité de recherche INRIA Futurs
Parc Club Orsay Université, ZAC des Vignes,
4, rue Jacques Monod, 91893 ORSAY Cedex (France)
Téléphone : +33 1 72 92 59 00 — Télécopie : +33 1 60 19 66 08

Heterogenous dating service with application to rumor spreading

Abstract: In this paper, we describe a fully decentralized algorithm, called "dating service" to organize communications into a fully heterogeneous network, that ensures that communication capabilities of the nodes are not exceeded. We prove that with high probability, this service ensures that a constant fraction of all possible communications is organized. Interestingly enough, this property holds true even if a node is not able to choose another node uniformly at random. We also present, as an application of the dating service, an algorithm for rumor spreading that enables to broadcast a unit-size message to all the nodes of a P2P system in logarithmic number of steps with high probability.

Key-words: randomized algorithms, distributed algorithms, rumor spreading, heterogeneous platforms

1 Introduction

In this paper, we describe a fully decentralized algorithm, called *dating service* to organize communications into a fully heterogeneous network, that ensures that communication capabilities of the nodes are not exceeded. The abstract purpose of the scheme is to randomly join demands and supplies of some resource of many nodes into couples. An example of application is rumor spreading in a heterogenous network, where demands and supplies are possibilities of receiving and sending messages, respectively.

We assume the usual setting for the rumor spreading (see for example [KSSV00]), that is a single node knows the rumor originally and wants to broadcast it to everyone else. The communication is organized in rounds and the algorithm only decides (usually in some random fashion) which nodes send messages to which nodes in each round. This allows for extensions such as rumors appearing in the network in course of time and also dynamics of the networks, also node failures. The rumor is assumed to have a unit size, i.e. it takes exactly one round to send it from a node to another node. In our scheme we use additional small bandwidth for maintenance.

As mentioned, communication capabilities of the set of participating nodes are heterogeneous. We assume that network interfaces are full-duplex and therefore that nodes can send and receive data simultaneously. We also assume that nodes can communicate with several nodes concurrently. This model corresponds to modern network techniques (packet switching) to handle concurrent communications. However, the rate with which a network interface sends and receives data cannot increase infinitely as the number of concurrent communications increases. The data transfer rate cannot exceed the hardware limitation of the network interface. Moreover, each network interface is associated to an incoming and an outgoing buffer to control communication rates. In order to take these limitations into account, each node i is associated to an incoming bandwidth $b^{in}(i)$ and an outgoing bandwidth $b^{out}(i)$, which means that node i is able to receive $b^{in}(i)$ unit size messages and to send (simultaneously) $b^{out}(i)$ unit size messages during each round. In order to consider nodes with very different capabilities, the ratios $\frac{\max_i b^{in}(i)}{\min_i b^{in}(i)}$ and $\frac{\max_i b^{out}(i)}{\min_i b^{out}(i)}$ are not a priori bounded. On the other hand, we assume that for a given node, the ratio between its incoming and outgoing bandwidths is bounded by a constant C :

$$\forall i, \quad \frac{1}{C} \leq \frac{b^{in}(i)}{b^{out}(i)} \leq C.$$

In order to organize communications, we rely on the dating service described in more details in Section 2. During the dating service phase, each node i sends to random nodes $b^{in}(i)$ message requests and $b^{out}(i)$ message offers. In turn, each node organizes matchings between the requests and the offers it has received. We prove that with high probability, this scheme enables to organize a linear number of communications among all possible communications, given the capabilities of the nodes. More precisely, if $B^{in} = \sum_i b^{in}(i)$ denotes the overall incoming bandwidth and $B^{out} = \sum_i b^{out}(i)$ denotes the overall outgoing bandwidth, then the dating service organizes with high probability αm communications,

where $m = \min(B^{in}, B^{out})$. This result holds true for any distribution of the choice of the random nodes responsible for organizing the dates, provided that the same distribution is used by all nodes and for both requests and offers. More specifically, we prove that if nodes are chosen uniformly at random, then $\alpha > 0.44$, that is, almost half of possible communications are indeed organized by the dating service. Moreover our result (a linear fraction of possible communications is achieved) holds even if choosing a node uniformly at random is not possible, which makes our algorithm more practical. We compare the results obtained by uniform strategies and strategies based on Distributed Hash Tables (DHTs) in Section 4.

As an illustrative example of the use and the effectiveness of the dating service, we consider rumor spreading in Section 3. In each round, the dating service is used to organize communications between nodes. In our model, we do not assume that nodes stop asking for messages (i.e. stop sending requests) once they have the message nor do not send messages (i.e. do not send offers) if they have nothing to say, in order to cope with dynamicity and to keep the protocol very simple. Nevertheless, even in this context, we prove that using our rumor spreading algorithm, all n nodes receive the message with high probability after $O(\log n)$ rounds. This result is comparable with those obtained by PUSH, PULL and PUSH and PULL algorithms which are analyzed in details for example in [KSSV00]. Here we only say shortly how these schemes work.

In each round each node chooses another node uniformly at random. In PUSH model the former sends an information to the latter if. In PULL model it is the other way around. In case of PUSH and PULL scheme, the nodes exchange information. Main results of [KSSV00] are not only bounding time to $O(\log n)$ but also bounding total communication cost to $O(n \cdot \log n)$. In our analysis we do not bound the communication cost but our spreading algorithm does not require the capability of choosing another node uniformly at random. These results are assessed by a set of simulations (for both uniform and DHT-based dating services) in Section 4. Moreover, we prove that if the message starts from a node with sufficiently large outgoing bandwidth (at least $\Omega(\log n)$), nodes with at least average bandwidth $\Omega(m/n)$ will receive the message after only $O(\log n/\log(m/n))$ rounds with high probability, which opens the way for hierarchical content distribution, where nodes receive different messages according to their communication capabilities. Finally, in Section 5 we discuss several other possible uses of the dating service.

1.1 Notations

Throughout this paper we use the well known notations of $O()$ and $\Omega()$, where $O(f(n))$ means at most $c \cdot f(n)$ for constant c and sufficiently large n and $\Omega(f(n))$ means at least $c \cdot f(n)$ for constant c and sufficiently large n.

Together with the above notations we also use the term *with high probability* (in short : whp) in the following way : a random variable $X(n)$ is bounded by $O(f(n)$ with high probability means that for any constant ℓ, $\Pr[X > c \cdot f(n)] \leq \frac{1}{n^\ell}$, where the constant c depends only on ℓ. The term for $\Omega()$ notation is defined similarly.

2 Dating service

In this section we introduce and describe the dating service, a tool used to organize the communication in each round. The dating service will need some overhead communication but these will be only small messages — typically one IP address in each message. If we use the dating service to organize rumor spreading in which we broadcast a long file, say a movie, this overhead does not matter at all. In this section we prove good properties of the dating service, too.

First, we describe the dating service in the model assumed in the usual rumor spreading setting, that is we assume that each node has a possibility of sending a (tiny) message to another node chosen at random. In each round, the communication is organized as shown in Algorithm 1.

Algorithm 1 Dating service for node i

```
send b^out(i) requests for sending to randomly chosen nodes
send b^in(i) requests for receiving to randomly chosen nodes
receive s sending and r receiving requests
q = min{s, r}
choose uniformly at random q requests of each type
produce a random perfect matching of the chosen requests
for each sending request do
  if the request is among q chosen sending requests then
    send the address of neighbor in random matching to the originator
  else
    send information that sending is not possible to the originator
for each received answer containing an address do
  send the actual message to the given address
```

Note that in the above description, it is not necessary that the random choice of nodes be uniform – only that all requests are sent using the same distribution. This randomness is a load-balancing factor; as an extreme case, sending all requests to a single node would result in a centralized scheme.

Denote the total incoming bandwidth of nodes in the network by B^{in} and the total outgoing bandwidth by B^{out}. Let $m = \min\{B^{in}, B^{out}\}$, that is the maximum number of dates that a centralized service would be able to organize in a round. First we show that in expectation our distributed scheme loses at most a constant fraction of messages compared to a centralized one.

Lemma 1. *Let $X_1, X_2, ..., X_n$ be random variables denoting the number of dates organized by nodes $1, ..., n$. Let $X = X_1 + ... + X_n$. Then the expected value $\mathrm{E}[X] = \Omega(m)$.*

Démonstration. We prove the lower bound assuming $B^{in} = B^{out} = m$; increasing the larger of the two can only result in more dates and a higher ratio $\mathrm{E}[X]/m$.

Let the probabilities with which nodes are chosen as destinations of requests be $p_1, ..., p_n$, where $p_1 + ... + p_n = 1$. Without loss of generality we assume $p_1 \geq p_2 \geq ... \geq p_n$. Call one such individual probability "small" if it is less than $1/2m$, and "large" otherwise. The sum of all "small" probabilities is at most $n/2m \leq 1/2$, so that the sum of "large" probabilities is at least $1/2$.

Now view each node i as a "big" bucket of size p_i, and consider it as a union of $\lfloor 4mp_i \rfloor$ sub-buckets, each of size exactly $1/4m$, plus one that is strictly smaller. Obviously, the number of dates obtained by considering these sub-buckets will be less than that produced by the bigger, union buckets. Now, each "large" probability cuts into at least two full buckets, so that we lose at most $1/3$ of its size by considering only full sub-buckets. In other words, the number of full $1/4m$-sized sub-buckets we obtain just from large probabilities is at least $4m/3$.

Now we only need to show that for a single of sub-bucket there are on expectation $\Omega(1)$ dates created solely by this bucket. A bucket has size exactly $\frac{1}{4m}$ and we have two independent random variables X and Y for the numbers of sending and receiving requests, respectively. Both X and Y are binomial $\mathrm{Bi}(m, \frac{1}{4m})$ with expectation $E[X] = E[Y] = \frac{1}{4}$. Now, it is a standard result of probability theory that binomial variables are well approximated by Poisson variables ; in our case, since $m \geq n$, we get from, say, [BHJ92, p2] that the total variation distance between the $\mathrm{Bi}(m, \frac{1}{4m})$ and $\mathrm{Po}(\frac{1}{4})$ distribution is $O(1/m)$. Thus, the expected number of dates produced by a sub-bucket is the probability that this number is positive, which is $p + O(1/m)$ where p is the probability that two independent $\mathrm{Po}(1/4)$ variables are both non-zero :

$$p = \left(1 - e^{-1/4}\right)^2 \simeq 0.048.$$

Summing over all sub-buckets, the expected number of dates in a given round is at least $0.064 \cdot m + \Omega(1)$, where the $\Omega(1)$ term is uniform with respect to the probability distribution $p_1, \ldots, p_n$. □

We comment here that the proof above is correct but definitely not optimal. Using the Poisson approximation in the uniform case $p_i = 1/n$, we get an estimate of $0.44 \cdot n$ when $m = n$, and the ratio $\mathrm{E}[X]/m$ is an increasing function of m/n. Furthermore, we conjecture that the uniform distribution is in fact the worst case for this ratio. That is, if some nodes have higher probability of being chosen, they attract more requests and arrange more dates. Our experiments in Section 4 confirm this, but we do not attempt to give proofs here.

We now show that the number of dates is linear not only on expectation but also with high probability.

Lemma 2. *Let X_i and X be defined as in Lemma 1. Then $X = \Omega(m)$, with high probability.*

Démonstration. We use the Independent Bounded Differences Inequality as described in [McD98, Theorem 3.1]. Again assuming that $B^{in} = B^{out} = m$, the total number X of dates is a function of $2m$ independent random variables (each variable indicating which node a given sending or receiving request is sent to), such that changing the value of a single random

variable (the destination of a single request) can change the total by at most one. Thus, by Theorem 3.1 in [McD98], the probability that X deviates from its expectation by more than t is at most $2e^{-t^2/m}$; taking $t = \Omega(m^{1/2+\epsilon})$ makes this probability exponentially small. □

We introduce a (universal) constant β such that we can say : with high probability $X \geq \beta \cdot m + \Omega(1)$. As said above, we believe that $\beta = 0.4$ is good but we only proved the statement for $\beta = 0.064$.

The construction of the dating service also gives us the following *symmetry property* of the set of dates produced by the dating service.

Lemma 3. *Consider the complete bipartite graph $G = K_{B^{out},B^{in}}$, with each node in the first (resp. second) vertex set corresponding to a unit of outgoing (resp. incoming) bandwidth for a given node.*

Then, conditionally on the total number of arranged dates being k, the set of dates itself is a uniform random k-matching of G.

Démonstration. We only have to check that two different k-matchings have the same probability of occurring. This is a simple consequence of the construction of the dating service, which is defined as a function of a set of independent, identically distributed choices (namely, destinations of requests and random selection of matching requests by each node). Consider two possible k-matchings m_1 and m_2, and some permutation σ of the vertex sets of G that changes m_1 into m_2. This σ naturally induces a measure-preserving transformation T of the set of random variables used to define the k-matchings, such that T transforms the event "m_1 is obtained" into the event "m_2 is obtained" (simply put, if the sending request corresponding to a left vertex u goes to some node v, then after T, the sending request corresponding to $\sigma(u)$ goes to the same v, with the same arrangement for receiving requests of right vertices ; and the matching choices are similarly rearranged). Thus, the two matchings m_1 and m_2 have the same probability. □

An important consequence of this lemma is the fact that each possible outgoing (resp. incoming) link produces a date with the same probability $p \geq \beta B^{out}/m$ (resp. $p' \geq \beta B^{in}/m$), even though the dates are not independent ; and, conditional on the total number of dates, the number of incoming (resp. outgoing) dates for any single node (or for any set of nodes) follows a hypergeometric distribution.

3 Rumor spreading

In this section we describe and analyze an application of the dating service to rumor spreading. The basic problem is as follows. In the beginning there is a single node which knows some information — the rumor. We are looking for a scheme which spreads the rumor in the whole network as fast as possible. PUSH and PULL algorithms mentioned in Section 1 do it in a number of rounds $O(\log n)$. We describe a scheme based on the dating service which achieves the same kind of bound, but makes efficient use of heterogeneous bandwidths by speeding up the process for large bandwidth nodes.

We remind that the network is heterogenous with restriction that the ratio between in- and outgoing bandwidth of each node is bounded by C. The rumor spreading scheme is given by the dating service algorithm. Namely it is the last step of the algorithm. We prove below that using such a scheme, all nodes will know the rumor after at most logarithmic (in n) number of rounds.

Theorem 4. *With high probability, the rumor spreading process based on our dating service sends the rumor to all nodes in $O(\log n)$ rounds.*

Démonstration. We split the analysis into three phases, depending on the total outgoing bandwidths of informed nodes ; we denote this bandwidth in round t by I_t :

1. from $I_0 \geq 1$ until $I_t = \Omega(\max(\frac{m}{n}, \log n))$;
2. from the end of phase 1 until $I_t \geq m/2$;
3. from the end of phase 2 until all nodes are informed.

In the following we show that each phase separetely takes $O(\log n)$ rounds. For the first two phases we will need the following definition and lemma :

Definition 5. *Fix a constant $\alpha < 1$. Take all I_t links outgoing from informed nodes and consider them in any fixed order. We say that a link is successful in round t, if (i) it succeeds in contacting a node not previously contacted (not even by any informed link already considered in round t) and (ii) the outgoing bandwidth of the contacted node is at least $\alpha \cdot \frac{m}{n}$*

For such a definition of success we now prove a lower bound on the probability that a given link is successful in a given round during phases 1 and 2.

Lemma 6. *Fix $\alpha = 1/4$ in the previous definition. Then, in every round t, under the condition that $I_{t+1} \leq \frac{m}{2}$, an outgoing link is successful with probability at least $\frac{\beta}{4C^2}$.*

Démonstration. In the reasoning below we condition on the number of dates in the current round being at least $\beta \cdot m$, which happens with high probability (by Lemma 2).

We consider two cases :

1. $m = B^{out} \leq B^{in}$
2. $m = B^{in} \leq B^{out}$

In the first case, the probability for an outgoing link of getting a date is at least β. Consider all uninformed nodes which have outgoing bandwidth at most $\alpha \cdot \frac{m}{n}$. Their total bandwidth is at most $\alpha \cdot m$, so the total uninformed outgoing bandwidth we would like to hit is at least $(1 - \frac{1}{2} - \alpha) \cdot m = \frac{m}{4}$. The total incoming bandwidth of these nodes is at least $\frac{m}{4C}$, whereas the total incoming bandwidth of all nodes is at most $C \cdot m$, so based on Lemma 3, the probability of hitting one of these "interesting" nodes is at least $\frac{1}{4C^2}$. The joint probability of getting a date and hitting a correct node is thus at least $\frac{\beta}{4C^2}$.

In the second case the reasoning is similar, except that an outgoing link has a probability of getting a date in round t at least $\frac{\beta}{C}$ and that the total incoming bandwidth is at most m. Thus, the resulting probability of success is also at least $\frac{\beta}{4C^2}$. □

Now we are able to bound the length of the first phase.

Lemma 7. *The first phase ends in at most $O(\log n)$ time steps, with high probability.*

Démonstration. We know that there is at least one outgoing link from a node which initially has the message. Each round of phase 1, this same link has a lower bounded probability of success ; since rounds are independent, the number of rounds until this link gets at least $d \cdot \log n$ successes is at most $d' \cdot \log n$, with high probability. □

For the second phase we show that in each round we multiply by a constant factor (strictly larger than 1) the number of informed outgoing links, with high probability. Since we start with at least $\frac{m}{n}$ such links, $O(\log_{(m/n)} n)$ rounds would then suffice to come to $\frac{m}{2}$ informed links. The precise statement is formulated as follows :

Lemma 8. *In each round of phase 2, with high probability, a constant fraction of connections succeeds. Phase 2 lasts $O(\log n / \log(1 + \frac{m}{n}))$ rounds.*

Démonstration. Let $I_t \geq \Omega(\log n)$ be the number of informed outgoing links in round t. Each of them succeeds with probability $p_t = \Omega(1)$ (though not independently), which exactly means that, on expectation, a constant fraction of them are successful. We again use the Independent Bounded Differences Inequality to prove that this constant fraction holds with high probability.

We start by conditioning on the total number of dates in the current round being k (again, this k is at least $\beta \cdot m$ with high probability). Then the set of matched outgoing links from informed nodes is independent of the set of matched incoming links to uninformed nodes, and the cardinality of the first set follows the hypergeometric distribution with parameters (k, B^{out}, I_t).

Dates can be faithfully simulated using a relatively low number of independent random variables, as follows : number all outgoing links 1 to B^{out} in an arbitrary order with the only condition that those from informed nodes are numbered 1 to I_t, and take k independent uniform random permutations $\sigma_1, \ldots, \sigma_k$ of $[[1, B^{out}]]$. Similarly, number all incoming links 1 to B^{in} in an arbitrary order, and take k independent uniform random permutations σ'_1, σ'_k of $[[1, B^{in}]]$.

Now the outgoing (resp. incoming) end of each date can be chosen as follows : the i-th matched link is the first element in permutation σ_i (resp. σ'_i) that is not among the $i-1$ first matched links.

The total number of successes in the round thus appears as a function of $2k$ independent random variables, and changing just one of these random variables can only result in changing at most one of the matched links (though it can result in changing an arbitrary number of matches, at most one unmatched link can become matched) ; as a consequence, changing one of the permutations can change the total number of successes by at most 1. Thus, we can apply the Independent Bounded Differences Inequality with constant $c = 1$, which proves

that, if $\mu = \mu(I_t, k)$ is the expected number of successes, S is the number of successes, and X is the number of dates, we get

$$\Pr(|S - \mu| \geq t | X = k) \leq 2e^{-t^2/k}$$

Taking $t = \sqrt{a \cdot k \cdot \log n}$ above yields an upper bound of $O(n^{-a})$. Since $k = \Omega(\log n)$ and we can choose the constant to be as large as we need it (by changing the constant in the $\Omega()$ in the definition of phase 1), this proves that, with high probability, the deviation of number of successes from its expectation is less than a constant fraction of its expectation.

The claim on the duration of phase 2 follows : with high probability, we have $\Omega(I_t)$ successes in each round t of phase 2, which means that $I_{t+1} \geq (1 + \gamma \frac{m}{n}) \cdot I_t$ (where $\gamma > 0$ is the constant – which we make no attempt to make explicit or optimize – in the "constant proportion of successes" claim) with high probability. Since phase 2 ends when I_t has been multiplied by a factor of

$$\frac{m/2}{\max\left(\frac{m}{n}, \log n\right)} = \min\left(n, \frac{m}{\log n}\right),$$

we get a high probability upper bound on the duration of phase 2 of

$$O\left(\frac{\log\left(\min(n, m/n)\right)}{\log(1 + m/n)}\right).$$

□

Finally, we bound the length of the last phase.

Lemma 9. *With high probability, it takes at most $O(\log n)$ rounds to send the rumor to all the uninformed nodes, if we start with $\Omega(m)$ outgoing connections belonging to informed nodes.*

Démonstration. Consider a single uninformed node, it has at least one incoming link. Reversing the roles of incoming and outgoing links from the proof of Lemma 7, we see that, each round, each such incoming link has constant probability of having a date with an outgoing link from an informed node ; thus, with high probability, $d \cdot \log n$ rounds suffice for such a date, for an appropriate constant d. Taking a union bound on the probabilities for each uninformed node to remain uninformed, we see that phase 3 ends in $O(\log n)$ rounds with high probability. □

Overall, Lemmas 7, 8, and 9, yield Theorem 4 □

We now come to the real strength of the dating service concerning heterogenous networks. Assume that the network is heterogenous and such that $m > n \log n$, i.e. the average bandwidth is larger than $\log n$ even though there may still be some weak nodes (with low bandwidth) in the network. If the rumor starts at a node with at least average bandwidth, that is at least $\Omega(m/n)$, then all other nodes of average bandwidth receive the rumor after

$O(\frac{\log n}{\log(1+m/n)})$, with high probability. This means time $O(\frac{\log n}{\log\log n})$ for average bandwidth $\Omega(\log n)$ and constant time for average bandwidth $\Omega(\sqrt{n})$.

Theorem 10. *Assume $m = \Omega(n \log n)$. If the node which has the rumor initially has bandwidth at least $\Omega(m/n)$, then all nodes with bandwidth $\Omega(m/n)$ receive the rumor within $O(\frac{\log n}{\log(m/n)})$ rounds, with high probability.*

Démonstration. Notice that Phase 1 is finished before any communication begins. Phase 2 lasts $O(\frac{\log n}{\log(1+m/n)})$ rounds, as shown in Lemma 8. Thus, we only need to show that after Phase 2 has finished, all average nodes receive the rumor within $O(\frac{\log n}{\log(1+m/n)})$ rounds.

Consider a single uninformed node with incoming degree $\Omega(\log n)$. Basing on the proof of Lemma 8 we know that a constant fraction of its requests for receiving are fulfilled with high probability. Each of the incoming messages contains the rumor with probability at least $1/2$, so in a single round such a node does not get the rumor with probability inversely polynomial in n. After constant number of rounds such a node is informed with high probability. □

Even if the rumor starts in a weak node, we still have some chance :

Corollary 11. *Assume again that $m = \Omega(n \log n)$. Independently of the bandwidth of the node on which the rumor starts, it takes on expectation $O(\frac{\log n}{\log(m/n)})$ rounds to deliver the rumor to all nodes with at least average bandwidths, provided that the average bandwidth is $m/n \geq \Omega(\log n)$.*

Démonstration. Just notice that the single node which knows the rumor initially in each round has constant probability of sending the message to an average node. This takes constant number of rounds on expectation. Then, with high probability all the average nodes are informed in $O(\frac{\log n}{\log(m/n)})$ rounds by Theorem 10 □

4 Practical considerations and simulation results

In practice we propose to use Distributed Hash Tables (see for example [RFH+01, SMK+01, NW03a, NW03b]) as a foundation for the dating service. Very shortly, nodes of the network are distributed randomly on $(0, 1]$ ring and each node is responsible for the interval from itself to its successor. Some communication structure is built so, that every node can send a message to a node responsible for any value $x \in (0, 1]$. If in our dating service we send requests to nodes responsible for values chosen uniformly at random from $(0, 1]$, we choose nodes with distribution far from uniform (some nodes have intervals of lengths $O(1/n^2)$, some have $\Omega(\log n/n)$) but with the same distribution for each node.

We do not want to consider aspects such as dynamics of the network, synchronisation of rounds, load balancing (especially in heterogenous case), as these are all topics for further research to make the dating service even more practical. We want to mention one important aspect though. Routing in DHTs takes time $\Theta(\log n)$ or close and since we use it in each round, it would mean that each round takes such time. One can use pipelining of dates, that

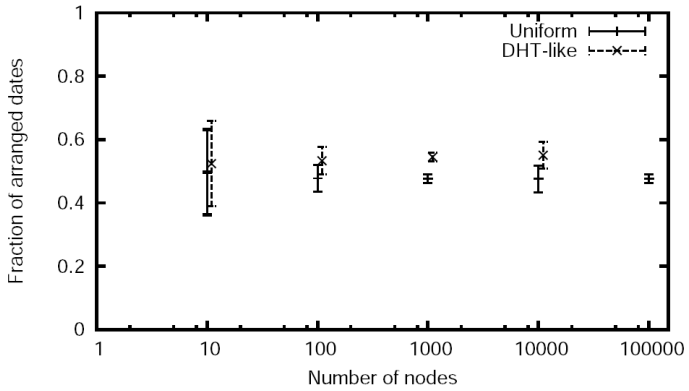

FIG. 1 – Fraction of dates arranged by dating service

is send requests for dates in each round even before receiving the answers for the previous one. Thus, after $\Theta(\log n)$ time steps, answers will start coming each round. This means that for k rounds of dating service we need time $\Theta(\log n + k)$.

We performed two kinds of experiments to show that our schemes are not only of theoretical interest, especially that the constant Lemma 1. One set of experiments was only in order to check, how many dates are generated in each round in two different settings : uniform and DHT-like. The second set of experiments concerns rumor spreading using the dating service.

Experiments for dating service

In the experiments meaning to check the actual number of dates in a round there were n nodes (for $n \in \{10, 10^2, 10^3, 10^4, 10^5\}$) and they generated n requests of each type. For uniform distribution of requests ($p_1 = ... = p_n$) we ran 10^4 or 10^3 rounds to calculate the average and standard deviation. The experiments were performed on a single computer — therefore for $n \in \{10^4, 10^5\}$) we were able to run only 10^3 rounds. The average number of arranged dates appears to be always slightly more than $0.47 \cdot n$.

To check how our process behaves in a DHT setting, we generated a DHT and then ran 10^4 or 10^3 rounds. Again we calculated averages and standard deviations. For the final result we took only one DHT out of 200 generated — the one that showed *the worst* average. The number of dates for these worst found DHTs are above $0.52 \cdot n$. For the best ones they reach as high as $0.67 \cdot n$ for $n = 10$ but are close to $0.55 \cdot n$ for the largest $n = 10^4$. In this case we were not able to run the experiments for $n = 10^5$.

The results are summarized in Figure 1.

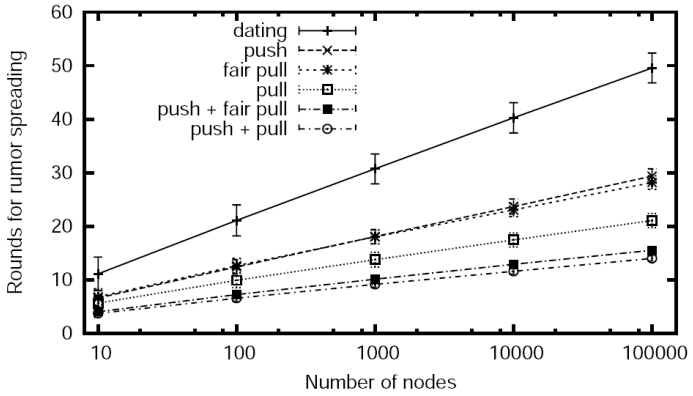

FIG. 2 – How many rounds it takes to spread a single rumor

Experiments for rumor spreading

In order to check how does our dating service behave in rumor spreading in comparison to other schemes we performed similar experiments. The number of nodes ranged from 10 to $100,000$ and we had the following algorithms : simple PUSH, simple PULL, simple PUSH and PULL, fair PULL — in which a node satisfies only one request when it is asked for information and fair PUSH and PULL. We compared them with rumor spreading based on dating service with uniform organization of dates. We did not perform experiments for DHT-like structures — from the previous set of experiments it follows that they will be at least as fast as rumor spreading based on uniform dating service. Again, in order to calculate averages and standard deviations, we performed each experiment 10^4 or 10^3 times.

The averages together with standard deviations are shown in Figure 2. We can see that the order of the algorithms from the best to the worst is as follows : PUSH and PULL, PUSH and fair PULL, PULL, fair PULL, PUSH, dating service. The following comment is needed here : notice that PUSH and PULL methods benefit from double communication in each round - one for PUSH and one for PULL. Also the unfair versions of the algorithms may benefit from much higher bandwidth — when many nodes PULL an informed node at the same time. Therefore we should actually compare the rumor spreading based on the dating service only with the PUSH and fair PULL methods. It is less than 2 times slower than them. We think that it is a fair price to pay for possibility of distributed implementation and always respecting bandwidths.

5 Future Works and possible extensions

We strongly believe that the dating service can be used as a building block for many distributed services where nodes offer and request for resources. The first possible extension

consist in considering the case of rumor mongering (or equivalently the broadcast of a large message). In this context, the message is split into smaller parts and is sent in a pipelined fashion through the network. In this case, we can make a deeper use of the dating service mechanism, since both incoming and outgoing bandwidths can be used efficiently. The most challenging problem consists in organizing the communications, so as to ensure that each part of the message is received exactly once. To achieve this goal, randomized network coding techniques [HeS+03] have proven their efficiency [DMC06].

The dating service may also be used in distributed replicated storage systems. In this context, each node offers room (in terms of block) to store remote objects and requests room to store remotely its local objects. In this case, the dating service may be used to organize block exchanges between nodes. As already stressed out, one of the main advantages of the dating service scheme is that it can be implemented other existing DHT-based systems, without requiring the ability to perform sophisticated requests such as the choice of a uniform random node.

6 Conclusion

In this paper, we describe a fully decentralized algorithm, called "dating service" to organize communications into a fully heterogeneous network, that ensures that communication capabilities of the nodes are not exceeded. We prove that with high probability, this service ensures that a constant fraction of all possible communications is organized. Interestingly enough, this property holds true even if a node is not able to choose another node uniformly at random, contrarily to what happens for Push, Pull and Push and Pull algorithms. As an illustration, we present a practical implementation of the dating service based on currently available DHT systems and we assess its results through extensive simulations. We also present, as an application of the dating service, an algorithm for rumor spreading that enables to broadcast a unit-size message to all the nodes of a P2P system in logarithmic number of steps with high probability, i.e. the same complexity as the best known algorithms for rumor spreading, without the practical limitations of these solutions due to the capability of choosing nodes uniformly at random. At last, we discuss some possible uses of the dating service for rumor mongering, content distribution and distributed storage.

Références

[BHJ92] A. D. Barbour, Lars Holst, and Svante Janson. *Poisson Approximation*, volume 2 of *Oxford Studies in Probability*. Oxford Science Publications, 1992.

[DMC06] S. Deb, M. Mdard, and C. Choute. Algebraic gossip : A network coding approach to optimal multiple rumor mongering. *IEEE ACM Transaction on networking*, pages 2486 – 2507, 2006.

[HeS+03] T. Ho, M. edard, J. Shi, M. Effros, and D. Karger. On randomized network coding. In T. Ho, M. Medard, J. Shi, M. Effros, and D. R. Karger, editors, *Proceedings*

of 41st Annual Allerton Conference on Communication, Control, and Computing, 2003.

[KSSV00] Richard M. Karp, Christian Schindelhauer, Scott Shenker, and Berthold Vocking. Randomized rumor spreading. In *IEEE Symposium on Foundations of Computer Science*, pages 565–574, 2000.

[McD98] Colin McDiarmid. *Concentration*, volume 16 of *Algorithms and Combinatorics*, pages 195–248. Springer, 1998.

[NW03a] Moni Naor and Udi Wieder. A Simple Fault Tolerant Distributed Hash Table. In *Proc. of the 2nd International Workshop on Peer-to-Peer Systems*, pages 88–97, 2003.

[NW03b] Moni Naor and Udi Wieder. Novel Architectures for P2P Applications : the Continuous-Discrete Approach. In *Proc. of the 15th ACM Symp. on Parallel Algorithms and Architectures (SPAA)*, pages 50–59, 2003.

[RFH+01] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard M. Karp, and Scott Shenker. A Scalable Content Addressable Network. In *Proc. of the ACM SIGCOMM*, pages 161–172, 2001.

[SMK+01] Ion Stoica, Robert Morris, David R. Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord : A Scalable Peer-to-Peer Lookup Service for Internet Applications. In *Proc. of the ACM SIGCOMM*, pages 149–160, 2001.

Unité de recherche INRIA Futurs
Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399